适用于 Ruby 的 AWS 开发工具包

开发人员指南

适用于 Ruby 的 AWS 开发工具包: 开发人员指南

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

AWS适用于 SDK for Ruby 是什么？ 1
其他文档和资源 1
部署到AWS云端 1
开发工具包主要版本的维护和支持 1
开始使用 2
使用 SDK 身份验证AWS 2
启动AWS访问门户会话 3
更多身份验证信息 3
安装 SDK 3
先决条件 3
安装 开发工具包 3
你好教程 4
编写代码 4
运行程序 5
Windows 用户的注意事项 5
后续步骤 6
与 SDK AWS Cloud9 一起使用 6
步骤 1：将您的设置AWS 账户为使用 AWS Cloud9 6
步骤 2：设置您的AWS Cloud9开发环境 6
步骤 3：为 Ruby 设置AWS开发工具 7
步骤 4：下载示例代码 7
步骤 5：运行示例代码 8
配置 SDK 9
凭证提供商链 9
创建AWS STS访问令牌 10
设置区域 10
使用共享config文件设置区域 10
使用环境变量设置区域 10
将区域设置为 Aws.config 11
在客户端或资源对象中设置区域 11
设置非标准端点 11
使用开发工具包 12
使用 REPL 实用程序 12
在 Ruby on Rails 上使用 SDK 12
调试提示：从客户端获取电线追踪信息 13
存根客户端的响应和错误 14
窃取客户的回应 14
存根客户端错误 15
分页 15
分页响应是可枚举的 15
手动处理分页响应 15
分页数据类 16
服务员 16
调用函数 16
等待函数 16
配置函数 17
延长函数 17
指定客户端超时时间 17
从 AWS SDK for Ruby 版本 1 或 2 迁移到版本 3 18
S ide-by-side 用法 18
一般差异 18
客户差异 18
资源差异 19
使用AWS服务 20

带有指导的代码示例 20
AWS CloudTrail 示例 20
亚马逊CloudWatch示例 25
AWS CodeBuild 示例 47
Amazon EC2 示例 49
AWS Elastic Beanstalk 示例 85
AWS Identity and Access Management(IAM) 示例 88
AWS KMS 示例 111
AWS Lambda 示例 113
Amazon Polly 示例 117
Amazon RDS 示例 120
Amazon S3 示例 124
Amazon SES 示例 165
Amazon SNS 示例 168
Amazon SQS 示例 171
亚马逊WorkDocs示例 190
代码示例 193
动作和场景 193
DynamoDB 193
EventBridge 210
AWS Glue 224
IAM 242
Lambda 261
Amazon RDS 271
Amazon S3 274
Amazon SNS 293
Amazon SQS 297
AWS STS 305
安全性 307
数据保护 307
Identity and Access Management 308
合规性验证 308
故障恢复能力 308
基础设施安全性 309
强制实施最低 TLS 版本 309
检查 OpenSSL 版本 309
升级 TLS 支持 309
S3 加密客户端迁移 310
迁移概述 310
更新现有客户端以读取新格式 310
将加密和解密客户端迁移到 V2 311
文档历史记录 314
..... cccxv

AWS适用于 SDK for Ruby 是什么？

欢迎使用AWS适用于 SDK for Ruby 开发人员指南。适用于 .by Ruby Ruby AWS Simple Storage Service (Amazon S3)、Amazon SElastic Compute Cloud (Amazon EC2) 和Amazon Dynam AWS 服务

《AWS SDK for Ruby 开发人员指南》提供了有关如何安装、设置和使用 AWS SDK for Ruby 来创建使用的 Ruby 应用程序的信息AWS 服务。

开始使用适用于 AWS SDK for Ruby (p. 2)

其他文档和资源

有关适用于 Ruby 开发人员的 AWS SDK 的更多资源，请参阅以下内容：

- AWSSDK 和工具参考指南 — 包含 SDK 中常见的设置、功能和其他基础概念 AWS
- AWS SDK for RubyAPI 参考-版本 3
- AWS代码示例存储库开启 GitHub
- RubyGems.org — 最新版本的 SDK 已模块化为特定服务的 gem，可在此处获取
  - 支持的服务-列出适用于 Ruby 的 AWS SDK 支持的所有 gem
- AWSSDK for Ruby 来源于GitHub：
  - 来源和自述文件
  - 每个 Gem 下的更改日志
  - 从 v2 迁移到 v3
  - 问题
  - 核心升级说明
- 开发人员博客
- Gitter 通道
- Twitter 上的 @awsforruby

部署到AWS云端

您可以使用AWS Elastic BeanstalkAWS OpsWorks、和AWS 服务等将应用程序部署AWS CodeDeploy到AWS云端。要使用 Elastic Beanstalk 部署 Ruby 应用程序，请参阅开发人员指南中的使用 EB CLI 和 Git 在 Ruby 中部署 Elastic Beanstalk 应用程序。AWS Elastic Beanstalk要使用部署 Ruby on Rails 应用程序AWS OpsWorks，请参阅将 Ruby on Rails 应用程序部署到AWSOpsWorks。有关AWS部署服务的概述，请参阅上的部署选项概述AWS。

开发工具包主要版本的维护和支持

有关维护和支持 SDK 主要版本及其基础依赖关系的信息，请参阅 AWS SDK 和工具参考指南中的以下内容：

- AWS开发工具和和和和和和和和和工具
- AWS开发工具和工具工具工具工具和工具工具和工具工具和工具

开始使用适用于 AWS SDK for Ruby

学习如何安装、设置和使用 SDK 创建 Ruby 应用程序以编程方式访问AWS资源。

主题

- 使用 SDK 身份验证AWS (p. 2)
- 安装AWS适用于 SDK for Ruby (p. 3)
- 适用于 Ruby 的AWS SDK 的你好教程 (p. 4)
- AWS Cloud9与 AWS SDK for Ruby 一起使用 (p. 6)

使用 SDK 身份验证AWS

在使用进行开发AWS时，必须确定如何对代码进行身份验证AWS 服务。您可以根据环境和可用的访问权限，以不同的方式配置对AWS资源的编程AWS访问权限。

要选择身份验证方法并针对 SDK 进行配置，请参阅 AWSSDK 和工具参考指南中的身份验证和访问。

我们建议在本地开发且雇主未向其提供身份验证方法的新用户进行设置AWS IAM Identity Center (successor to AWS Single Sign-On)。此方法包括AWS CLI为了便于配置和定期登录AWS访问门户而安装。如果您选择此方法，则在您完成《软件开发工具包和AWS工具参考指南》中的 IAM Id entity Center 身份验证程序后，您的环境应包含以下元素：

- AWS CLI，用于在运行应用程序之前启动AWS访问门户会话。
- 一种共享AWSconfig文件，[default]其配置文件包含一组可从 SDK 中引用的配置值。要查找此文件的位置，请参阅 AWSSDK 和工具参考指南中的共享文件的位置。
- 共享config文件设置region设置。这设置了 SDK 用于AWS请求的默认值AWS 区域。此区域用于未指定要使用的区域的 SDK 服务请求。
- 在向发送请求之前，SDK 使用配置文件的 SSO 令牌提供者配置来获取证书AWS。该sso_role_name值是连接到 IAM Identity Center 权限集的 IAM 角色，允许访问您的应用程序中AWS 服务使用的权限。

  以下示例config文件显示了使用 SSO 令牌提供者配置设置的默认配置文件。配置文件的sso_session设置指的是命名sso-session部分。该sso-session部分包含启动AWS访问门户会话的设置。

```
[default]
sso_session = my-sso
sso_account_id = 111122223333
sso_role_name = SampleRole
region = us-east-1
output = json

[sso-session my-sso]
sso_region = us-east-1
sso_start_url = https://provided-domain.awsapps.com/start
sso_registration_scopes = sso:account:access
```

AWSSDK for Ruby 无需向您的应用程序添加其他软件包（例如SSO和SSOOIDC）即可使用 IAM Identity Center 身份验证。

启动AWS访问门户会话

在运行可访问的应用程序之前，您需要一个活跃的AWS访问AWS 服务门户会话才能让 SDK 使用 IAM Identity Center 身份验证来解析证书。根据您配置的会话长度，您的访问权限最终将过期，SDK 将遇到身份验证错误。要登录AWS访问门户，请在中运行以下命令AWS CLI。

```
aws sso login
```

如果您遵循指南并设置了默认配置文件，则无需调用带有`--profile`选项的命令。如果您的 SSO 令牌提供者配置使用的是命名配置文件，则命令为`aws sso login --profile named-profile`。

要选择性地测试您是否已经有活动会话，请运行以下AWS CLI命令。

```
aws sts get-caller-identity
```

如果您的会话处于活动状态，则对此命令的响应会报告共享`config`文件中配置的 IAM Identity Center 账户和权限集。

> Note
>
> 如果您已经有一个活跃的AWS访问门户会话并正在运行`aws sso login`，则无需提供证书。登录过程可能会提示您允许AWS CLI访问您的数据。由于基AWS CLI于 SDK for Python 而构建，因此权限消息可能包含`botocore`名称的变体。

更多身份验证信息

人类用户，也称为人类身份，是应用程序的人员、管理员、开发人员、操作员和使用者。他们必须有身份才能访问您的 AWS 环境和应用程序。作为组织成员的人类用户称为员工身份。

访问时使用临时证书AWS。您可以使用身份提供商来以担任角色的形式提供为人类用户对 AWS 账户的联合访问权限，这将提供临时证书。对于集中式访问权限管理，我们建议使用AWS IAM Identity Center (successor to AWS Single Sign-On)（IAM Identity Center）来管理对您账户的访问权限以及这些账户中的其他权限。有关更多替代方案，请参阅以下：

- 要了解有关最佳实践的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。
- 要创建短期AWS证书，请参阅 IAM 用户指南中的临时安全证书。
- 要了解其他AWS SDK for Ruby 凭证提供商，请参阅 AWSSDK 和工具参考指南中的标准化证书提供商。

安装AWS适用于 SDK for Ruby

本节包括AWS适用于 SDK for Ruby 的先决条件和安装说明。

先决条件

在使用AWS适用于 SDK for Ruby 之前，必须通过进行身份验证AWS。有关设置身份验证的信息，请参阅使用 SDK 身份验证AWS (p. 2)。

安装 开发工具包

你可以像安装任何 Ruby gem 一样安装适用于 Ruby 的 AWS SDK。宝石可在以下网址获得RubyGems。AWS适用于 Ruby 的 SDK 采用模块化设计，由以下部分隔开AWS 服务。安装整`aws-sdk`颗宝石很大，可能需要一个多小时。

我们建议只安装宝石供AWS 服务你使用。它们的命名方式类似aws-sdk-*service_abbreviation*，完整列表可在 AWS SDK for Ruby README 文件的 “支持的服务” 表中找到。例如，与 Amazon S3 服务接口的 gem 可直接在上aws-sdk-s3获得。

Ruby 版本管理器

我们建议不要使用系统 Ruby，而是使用如下所示的 Ruby 版本管理器：

- RVM
- chruby
- rbenv

例如，如果您使用的是 Amazon Linux 2 操作系统，则可以使用以下命令更新 RVM，列出可用的 Ruby 版本，然后选择要使用适用于 R AWS SDK for Ruby 开发的版本。所需的最低 Ruby 版本为 2.3。

```
$ rvm get head
$ rvm list known
$ rvm install ruby-3.1.3
$ rvm --default use 3.1.3
```

捆绑器

如果您使用 Bundler，则以下命令会安装适用于 Amazon S3 AWS SDK for Ruby gem 软件开发工具包：

1. 安装捆绑器并创建：Gemfile

   ```
   $ gem install bundler
   $ bundle init
   ```

2. 打开已创建Gemfile的并为您的代码将要使用的每个AWS服务 gem 添加一gem行。要对 Amazon S3 示例进行操作，将以下一行添加到文件：

   ```
   gem "aws-sdk-s3"
   ```

3. 保存 Gemfile。
4. 安装在以下文件中指定的依赖关系Gemfile：

   ```
   $ bundle install
   ```

适用于 Ruby 的AWS SDK 的你好教程

使用适用于 SAWS DK for Ruby 向Amazon S3 问好。以下示例显示您的 Amazon S3 存储桶列表。

编写代码

将以下代码复制并粘贴到新的源文件中。将文件命名为 hello-s3.rb。

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource
```

```
  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
    false
  end
end

def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

AWS适用于 Ruby 的 SDK 采用模块化设计，由以下部分隔开AWS 服务。安装 Gem 后，Ruby 源文件顶部的`require`语句会导入 Amazon S3 服务的 SAWS DK 类和方法。有关可用AWS服务 gem 的完整列表，请参阅AWS SDK for Ruby README 文件的 “支持的服务” 表。

```
require 'aws-sdk-s3'
```

运行程序

打开命令提示符运行 Ruby 程序。运行 Ruby 程序的典型命令语法是：

```
ruby [source filename] [arguments...]
```

此示例代码不使用任何参数。要运行此代码，请在命令提示符下输入以下内容：

```
$ ruby hello-s3.rb
```

Windows 用户的注意事项

当您在 Windows 上使用 SSL 证书并运行 Ruby 编码时，您可能会看到类似于以下内容的错误。

```
C:\Ruby>ruby buckets.rb
C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `connect': SSL_connect returned=1
 errno=0 state=SSLv3 read server certificate B: certificate verify failed
 (Seahorse::Client::NetworkingError)
         from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `block in connect'

         from C:/Ruby200-x64/lib/ruby/2.0.0/timeout.rb:66:in `timeout'
         from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:921:in `connect'
         from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:862:in `do_start'
```

```
        from C:/Ruby200-x64/lib/ruby/2.0.0/net/http.rb:857:in `start'
...
```

要修复此问题，请在第一次AWS调用之前的某个位置将以下行添加到您的 Ruby 源文件中。

```
Aws.use_bundled_cert!
```

如果您在 Ruby 程序中只使用`aws-sdk-s3` gem 并且想要使用捆绑证书，则还需要添加`aws-sdk-core` gem。

后续步骤

要测试许多其他 Amazon S3 操作，请查看上的AWS代码示例存储库GitHub。

AWS Cloud9与 AWS SDK for Ruby 一起使用

AWS Cloud9是一种基于云的集成开发环境（IDE），包含一组工具，可用于在云中编写、构建、运行、测试、调试和发布软件。你可以AWS Cloud9与 AWS SDK for Ruby 一起使用，通过浏览器编写和运行 Ruby 代码。 AWS Cloud9包括代码编辑器和终端等工具。由于 AWS Cloud9 IDE 是基于云的，因此您可以使用联网的计算机在办公室、家中或任何地方处理项目。有关的一般信息AWS Cloud9，请参阅《AWS Cloud9用户指南》。

按照以下说明使用适用于 Rub AWS Cloud9 y 的 AWS SDK 进行设置：

- 步骤 1：将您的设置AWS 账户为使用 AWS Cloud9 (p. 6)
- 步骤 2：设置您的AWS Cloud9开发环境 (p. 6)
- 步骤 3：为 Ruby 设置AWS开发工具 (p. 7)
- 步骤 4：下载示例代码 (p. 7)
- 步骤 5：运行示例代码 (p. 8)

步骤 1：将您的设置AWS 账户为使用 AWS Cloud9

要使用AWS Cloud9，请从中登录到AWS Cloud9控制台AWS Management Console。

> Note
>
> 如果您使用AWS IAM Identity Center (successor to AWS Single Sign-On)进行身份验证，则可能需要将所需的权限添加`iam:ListInstanceProfilesForRole`到 IAM 控制台中的用户附加策略。

要在您的AWS账户中设置 IAM 实体以访问AWS Cloud9和登录AWS Cloud9控制台，请参阅AWS Cloud9用户指南AWS Cloud9中的团队设置。

步骤 2：设置您的AWS Cloud9开发环境

登录 AWS Cloud9 控制台后，请使用控制台创建 AWS Cloud9 开发环境。创建环境后，AWS Cloud9 会打开该环境的 IDE。

有关详细信息，请参阅《AWS Cloud9用户指南》AWS Cloud9中的创建环境。

> Note
>
> 在控制台中首次创建环境之后，我们建议您选择 Create a new instance for environment (EC2) (创建新的环境实例 (EC2))。此选项AWS Cloud9指示创建环境，启动 Amazon EC2 实例，然后将新实例连接到这个新环境。这是开始使用 AWS Cloud9 的最快方式。

如果终端未在 IDE 中打开，请打开它。在 IDE 中的菜单栏上，选择 Window, New Terminal (窗口、新终端)。您可以使用终端窗口安装工具和构建应用程序。

步骤 3：为 Ruby 设置AWS开发工具

AWS Cloud9打开开发环境的 IDE 后，使用终端窗口在您的环境中设置AWS适用于 Ruby 的 SDK。

你可以像安装任何 Ruby gem 一样安装适用于 Ruby 的 AWS SDK。宝石可在以下网址获得RubyGems。AWS适用于 Ruby 的 SDK 采用模块化设计，由以下部分隔开AWS 服务。安装整`aws-sdk`颗宝石很大，可能需要一个多小时。

我们建议只安装宝石供AWS 服务你使用。它们的命名方式类似`aws-sdk-`*`service_abbreviation`*，完整列表可在 AWS SDK for Ruby README 文件的 “支持的服务” 表中找到。例如，与 Amazon S3 服务接口的 gem 可直接在上`aws-sdk-s3`获得。

Ruby 版本管理器

我们建议不要使用系统 Ruby，而是使用如下所示的 Ruby 版本管理器：

- RVM
- chruby
- rbenv

例如，如果您使用的是 Amazon Linux 2 操作系统，则可以使用以下命令更新 RVM，列出可用的 Ruby 版本，然后选择要使用适用于 R AWS SDK for Ruby 开发的版本。所需的最低 Ruby 版本为 2.3。

```
$ rvm get head
$ rvm list known
$ rvm install ruby-3.1.3
$ rvm --default use 3.1.3
```

捆绑器

如果您使用 Bundler，则以下命令会安装适用于 Amazon S3 AWS SDK for Ruby gem 软件开发工具包：

1. 安装捆绑器并创建：`Gemfile`

   ```
   $ gem install bundler
   $ bundle init
   ```

2. 打开已创建`Gemfile`的并为您的代码将要使用的每个AWS服务 gem 添加一`gem`行。要使用 Amazon S3 示例，请在文件底部添加以下一行：

   ```
   gem "aws-sdk-s3"
   ```

3. 保存 Gemfile。
4. 安装在以下文件中指定的依赖关系`Gemfile`：

   ```
   $ bundle install
   ```

步骤 4：下载示例代码

使用终端窗口将适用于 Ruby 的 AWS SDK 的示例代码下载到AWS Cloud9开发环境中。

要将官方 AWS SDK 文档中使用的所有代码示例的副本下载到环境的根目录中，请运行以下命令：

```
$ git clone https://github.com/awsdocs/aws-doc-sdk-examples.git
```

适用于 Ruby 的 AWS SDK 的代码示例位于`ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby`目录中，其中`ENVIRONMENT_NAME`是您的开发环境的名称。

要继续使用 Amazon S3 示例，我们建议从代码示例开始`ENVIRONMENT_NAME/aws-doc-sdk-examples/ruby/example_code/s3/bucket_list.rb`。使用终端窗口导航到该`s3`目录并列出文件。

```
$ cd aws-doc-sdk-examples/ruby/example_code/s3
$ ls
```

要打开文件AWS Cloud9，可以`bucket_list.rb`直接在终端窗口中单击。

有关理解代码示例的更多支持，请参阅 AWSSDK for Ruby 代码示例 (p. 20)。

步骤 5：运行示例代码

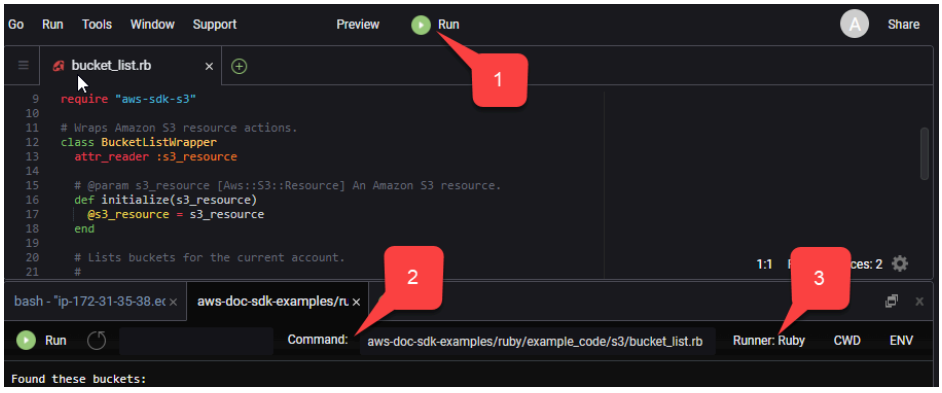
要在AWS Cloud9开发环境中运行代码，请选择顶部菜单栏中的 “运行” 按钮。 AWS Cloud9将自动检测`.rb`文件扩展名并使用 Ruby 运行器运行代码。有关在中运行代码的更多信息AWS Cloud9，请参阅《AWS Cloud9用户指南》中的 “运行您的代码”。

在下面的屏幕截图中，请注意以下基本区域：

- 1：运行。运行按钮位于顶部菜单栏上。这将为您的结果打开一个新选项卡。

  Note

  您还可以手动创建新的运行配置。在菜单栏上依次选择 Run（运行）> Run Configurations（运行配置）> New Run Configuration（新建运行配置）。
- 2：命令。 AWS Cloud9使用您运行的文件的路径和文件名填充命令文本框。如果您的代码需要传入任何命令行参数，则可以将这些参数添加到命令行中，方法与通过终端窗口运行代码时相同。
- 3：跑步者。 AWS Cloud9检测到您的文件扩展名是否为，`.rb`然后选择 Ruby Runner 来运行您的代码。



从运行代码生成的任何输出都显示在选项卡中。

配置 AWS SDK for Ruby

了解如何配置AWS适用于 SDK for Ruby。在使用进行开发AWS时，必须确定代码如何使用进行身份验证。AWS 服务您还必须设置AWS 区域要使用的

凭证提供商链

所有 SDK 都有一系列地点（或来源）供其检查，以获取用于向... 发出请求的有效凭证。AWS 服务找到有效凭据后，搜索即停止。这种系统搜索被称为默认凭证提供程序链进行搜索。

对于链中的每个步骤，都有不同的设置值的方法。直接在代码中设置值始终优先，然后设置为环境变量，然后在共享AWSconfig文件中设置。有关更多信息，请参阅《AWSSDK 和工具参考指南》中的设置优先顺序。

S AWSDK 和工具参考指南包含有关所有 SDK 使用的 S AWS DK 配置设置的信息，以及。AWS CLI要详细了解如何通过共享AWSconfig文件配置 SDK，请参阅共享配置和凭证文件。要详细了解如何通过设置环境变量来配置 SDK，请参阅环境变量支持。

要进行身份验证AWS，AWS适用于 Ruby 的 SDK 会按下表中列出的顺序检查证书提供者。

按优先顺序排列的凭证提供者	AWSSDK 和工具参考指南	AWS SDK for Ruby API 参考
静态凭证	静态证书	Aws::CredentialsAws::SharedCredentia
来自 AWS Security Token Service (AWS STS) 的 Web 身份令牌	假设角色凭证提供者 使用role_arnrole_session_name、和web_identity_token_file	Aws::AssumeRoleWebIdentityCredential
AWS IAM Identity Center (successor to AWS Single Sign-On)。 在本指南中，请参见使用 SDK 身份验证AWS (p. 2)。	IAM 身份中心证书提供商	Aws::SSOCredentials
可信实体提供商（例如AWS_ROLE_ARN）。在本指南中，请参见创建AWS STS访问令牌 (p. 10)。	假设角色凭证提供者 使用role_arn和role_session_name	Aws::AssumeRoleCredentials
流程凭证凭证提供程序	流程凭证提供者	Aws::ProcessCredentials
Azon Elastic Container Service (Amazon ECS) ice	容器凭证提供商	Aws::ECSCredentials
Azon Elastic Compute Cloud (Amazon EC2) 实例配置文件提供程序（IMDS 凭证提供程序）	IMDS 凭证提供商	Aws::InstanceProfileCredentials

如果设置了 AWS SDK f AWS_SDK_CONFIG_OPT_OUT or Ruby 环境变量，则不会解析共享AWSconfig文件（通常位 ~/.aws/config于）以获取证书。

如果您遵循推荐的新用户入门方法，则可以在入门主题使用 SDK 身份验证AWS (p. 2)中设置AWS IAM Identity Center (successor to AWS Single Sign-On)身份验证。其他身份验证方法对不同的情况很有用。为避免安全风险，我们建议始终使用短期凭证凭证。有关其他身份验证方法程序，请参阅 AWSSDK 和工具参考指南中的身份验证和访问权限。

创建AWS STS访问令牌

假设角色涉及使用一组可用来访问您通常无法访问的AWS资源的临时安全凭证。这些临时凭证由访问密钥 ID、秘密访问密钥和安全令牌组成。您可以使用该`Aws::AssumeRoleCredentials`方法创建 AWS Security Token Service (AWS STS) 访问令牌。

以下示例使用访问令牌创建 Amazon S3 客户端对象，其中`linked::account::arn`是要代入的角色的 Amazon 资源名称 (ARN)，`session-name`是代入角色会话的标识符。

```
role_credentials = Aws::AssumeRoleCredentials.new(
  client: Aws::STS::Client.new,
  role_arn: "linked::account::arn",
  role_session_name: "session-name"
)

s3 = Aws::S3::Client.new(credentials: role_credentials)
```

有关设置`role_arn`或或`role_session_name`改用共享AWSconfig文件进行设置的更多信息，请参阅 AWSSDK 和工具参考指南中的假设角色凭证提供者。

设置区域

使用大多数时需要设置区域AWS 服务。AWS适用于 Ruby 的 SDK 按以下顺序搜索区域：

1. 在客户端或资源对象中设置区域 (p. 11)
2. 使用设置区域 `Aws.config` (p. 11)
3. 使用环境变量设置区域 (p. 10)
4. 使用共享AWSconfig文件设置区域 (p. 10)

有关该`region`设置的更多信息，请参阅 AWS 区域AWSSDK 和工具参考指南。本节的其余部分从最常见的方法开始，介绍如何设置区域。

使用共享config文件设置区域

通过在共享AWSconfig文件中设置`region`变量来设置区域。有关共享`config`文件的更多信息，请参阅 AWSSDK 和工具参考指南中的共享配置和凭证文件。

在`config`文件中设置此值的示例：

```
[default]
region = us-west-2
```

如果设置了环境变量`AWS_SDK_CONFIG_OPT_OUT`，则不检查共享`config`文件。

使用环境变量设置区域

通过设置`AWS_REGION`环境变量来设置区域。

使用export命令在基于 UNIX 的系统（例如 Linux 或 macOS）上设置此变量。以下示例将区域设置为us-west-2。

```
export AWS_REGION=us-west-2
```

要在 Windows 上设置此变量，请使用 set 命令。以下示例将区域设置为us-west-2。

```
set AWS_REGION=us-west-2
```

将区域设置为 Aws.config

通过向Aws.config哈希region值添加值来设置区域。以下示例更新Aws.config哈希值以使用us-west-1区域。

```
Aws.config.update({region: 'us-west-1'})
```

您稍后创建的任何客户端或资源都绑定到该区域。

在客户端或资源对象中设置区域

创建AWS客户机或资源时设置区域。以下示例在us-west-1区域中创建 Amazon S3 资源对象。

```
s3 = Aws::S3::Resource.new(region: 'us-west-1')
```

设置非标准端点

该区域用于构造用于AWS请求的 SSL 终端节点。如果您需要在所选区域中使用非标准终端节点，请将endpoint条目添加到Aws.config。或者，在创建服务客户端或资源对象endpoint:时设置。以下示例在other_endpoint终端节点中创建 Amazon S3 资源对象。

```
s3 = Aws::S3::Resource.new(endpoint: other_endpoint)
```

使用AWS适用于 SDK for Ruby

本节提供有关使用AWS适用于 Ruby 的 SDK 开发软件的信息，包括如何使用 SDK 的某些高级功能。

S AWSDK 和工具参考指南还包含许多 SDK 中常见的设置、功能和其他基础概念。AWS

主题

- 使用AWS适用于 SDK for Ruby REPL 实用程序 (p. 12)
- 在 Ruby on Rails 上使用 SDK (p. 12)
- 调试提示：从客户端获取电线追踪信息 (p. 13)
- 存根客户端的响应和错误 (p. 14)
- 分页 (p. 15)
- 服务员 (p. 16)
- 指定客户端超时时间 (p. 17)
- 从 AWS SDK for Ruby 版本 1 或 2 迁移到版本 3 (p. 18)

使用AWS适用于 SDK for Ruby REPL 实用程序

这款 ge `aws-sdk` m 包括一个 Read-Eval-Print-Loop (REPL) 交互式控制台实用程序，它提供了一种测试适用于 SDK for Ruby 简单方法。Ruby 宝石的 SDK 可在 RubyGems.org 上找到。有关安装 SDK 的更多信息，请参阅安装AWS适用于 SDK for Ruby (p. 3)。

尽管该程序确实可以与 Interactive Ruby Shell (`irb`) 配合使用，但我们建议你安装`pry`，它提供了更强大的 REPL 环境。

使用以下命令来安装 `pry`。

```
gem install pry
```

您可以通过`aws-v3.rb`从命令行运行来访问程序。

```
aws-v3.rb
```

或者，您可以通过设置详细标志来启用 HTTP 线路日志记录。HTTP 线路日志记录提供有关适用于 Ruby 的 AWS SDK 与之间的通信的信息AWS。请注意，verbose 标志还会增加开销，从而使您的代码运行速度变慢。

```
aws-v3.rb -v
```

SDK for Ruby 包含为提供接口的客户端类AWS 服务。每个客户端类都支持一个特定AWS 服务的。在 REPL 中，每个服务类都有一个帮助器，它返回一个用于与该服务交互的新客户端对象。助手的名称将是转换为小写的服务名称。例如，Amazon S3 和 Amazon EC2 帮助对象的名称分别是`s3`和`ec2`。

在 Ruby on Rails 上使用 SDK

Ruby on Rails 提供了一个 Web 开发框架，利用它，可轻松地使用 Ruby 创建网站。

AWS为轻松与 Rails 集成提供了aws-sdk-rails宝石。您可以使用AWS Elastic Beanstalk、AWS OpsWorksAWS CodeDeploy、或 AWSRails 配置器在AWS云中部署和运行 Rails 应用程序。

有关安装和使用 ge aws-sdk-rails m 的信息，请参阅GitHub存储库 https://github.com/aws/ aws-sdk-rails。

调试提示：从客户端获取电线追踪信息

您可以通过设置http_wire_trace布尔值从AWS客户端获取电线追踪信息。电汇跟踪信息有助于区分客户端更改、服务问题和用户错误。什么时候true，设置会显示线路上正在发送的内容。以下示例创建了一个在创建客户端时启用了线路追踪的 Amazon S3 客户端。

```
s3 = Aws::S3::Client.new(http_wire_trace: true)
```

鉴于以下代码和参数 bucket_name，输出会显示一条消息来表明是否存在该名称的存储桶。

```
require 'aws-sdk-s3'

s3 = Aws::S3::Resource.new(client: Aws::S3::Client.new(http_wire_trace: true))

if s3.bucket(ARGV[0]).exists?
  puts "Bucket #{ARGV[0]} exists"
else
  puts "Bucket #{ARGV[0]} does not exist"
end
```

输出类似于以下内容。(为便于阅读，向 HEAD 行添加了回车。)

```
opening connection to bucket_name.s3-us-west-1.amazonaws.com:443...
opened
starting SSL for bucket_name.s3-us-west-1.amazonaws.com:443...
SSL established, protocol: TLSv1.2, cipher: ECDHE-RSA-AES128-GCM-SHA256
-> "HEAD / HTTP/1.1
    Accept-Encoding:
    User-Agent: aws-sdk-ruby3/3.171.0 ruby/3.2.2 x86_64-linux aws-sdk-s3/1.120.0
    Host: bucket_name.s3-us-west-1.amazonaws.com
    X-Amz-Date: 20230427T143146Z
/* omitted */
Accept: */*\r\n\r\n"
-> "HTTP/1.1 200 OK\r\n"
-> "x-amz-id-2: XxB2J+kpHgTjmMUwpkUI1EjaFSPxAjWRgkn/+z7YwWc/
iAX5E3OXRBzJ37cfc8T4D7ELC1KFELM=\r\n"
-> "x-amz-request-id: 5MD4APQQS815QVBR\r\n"
-> "Date: Thu, 27 Apr 2023 14:31:47 GMT\r\n"
-> "x-amz-bucket-region: us-east-1\r\n"
-> "x-amz-access-point-alias: false\r\n"
-> "Content-Type: application/xml\r\n"
-> "Server: AmazonS3\r\n"
-> "\r\n"
Conn keep-alive
Bucket bucket_name exists
```

您还可以在创建客户端后开启线路追踪。

```
s3 = Aws::S3::Client.new
s3.config.http_wire_trace = true
```

有关所报告的电线追踪信息中字段的更多信息，请参阅 Transfer F amily 必填请求标头。

存根客户端的响应和错误

了解如何在 AWS SDK for Ruby 应用程序中存根客户端响应和客户端错误。

窃取客户的回应

当你存根响应时，AWSSDK for Ruby 会禁用网络流量，客户端会返回存根（或虚假）的数据。如果您未提供被清除的数据，则客户端会返回：

- 作为空数组的列表
- 作为空哈希的映射
- 数值 0
- 日期 now

以下示例返回 Amazon S3 存储桶列表的存根名称。

```
require 'aws-sdk'

s3 = Aws::S3::Client.new(stub_responses: true)

bucket_data = s3.stub_data(:list_buckets, :buckets => [{name:'aws-sdk'}, {name:'aws-
sdk2'}])
s3.stub_responses(:list_buckets, bucket_data)
bucket_names = s3.list_buckets.buckets.map(&:name)

# List each bucket by name
bucket_names.each do |name|
  puts name
end
```

运行此代码将显示以下内容。

```
aws-sdk
aws-sdk2
```

> **Note**
>
> 在您提供任何被清除的数据后，默认值将不再应用于任何剩余的实例属性。这意味着在上一个示例中，剩余实例属性 `creation_date` 不是 now，而是 `nil`。

S AWS DK for Ruby 会验证您的存根数据。如果您传入的数据的类型错误，则会引发 `ArgumentError` 异常。例如，如果您使用以下内容，而不是 `bucket_data` 以前的分配：

```
bucket_data = s3.stub_data(:list_buckets, buckets:['aws-sdk', 'aws-sdk2'])
```

AWS适用于 SDK for Ruby 引发了两个`ArgumentError`异常。

```
expected params[:buckets][0] to be a hash
expected params[:buckets][1] to be a hash
```

存根客户端错误

你也可以存根适用于 Ruby 的 AWS SDK 针对特定方法引发的错误。以下示例显示了 `Caught Timeout::Error error calling head_bucket on aws-sdk`。

```
require 'aws-sdk'

s3 = Aws::S3::Client.new(stub_responses: true)
s3.stub_responses(:head_bucket, Timeout::Error)

begin
  s3.head_bucket({bucket: 'aws-sdk'})
rescue Exception => ex
  puts "Caught #{ex.class} error calling 'head_bucket' on 'aws-sdk'"
end
```

分页

有些AWS调用提供分页响应，以限制每次响应返回的数据量。一页数据表示多达 1,000 个项目。

分页响应是可枚举的

处理分页响应数据的最简单方法是使用响应对象中的内置枚举器，如以下示例所示。

```
s3 = Aws::S3::Client.new

s3.list_objects(bucket:'aws-sdk').each do |response|
  puts response.contents.map(&:key)
end
```

这为所进行的每个 API 调用生成一个响应对象，并枚举指定存储桶中的对象。开发工具包检索其他数据页面以完成请求。

手动处理分页响应

要自己处理分页，请使用响应的 `next_page?` 方法来验证是否有其他要检索的更多页面，或使用 `last_page?` 方法来验证是否没有其他要检索的页面。

如果有其他页面，请使用 `next_page` (请注意没有 ?) 方法来检索下一页结果，如以下示例所示。

```
s3 = Aws::S3::Client.new

# Get the first page of data
response = s3.list_objects(bucket:'aws-sdk')

# Get additional pages
while response.next_page? do
  response = response.next_page
  # Use the response data here...
end
```

> **Note**
>
> 如果您调用该`next_page`方法但没有其他页面可供检索，则 SDK 会引发 Aws:PageableResponse:: LastPageError 异常。

分页数据类

适用于 Ruby 的 AWS SDK 中的分页数据由 Aws:: 类处理，该PageableResponse类包含在 Seahorse:: Client:: Response 中，用于提供对分页数据的访问权限。

服务员

Waiter 是实用程序方法，用于轮询客户端上出现的特定状态。按为服务客户端定义的轮询间隔尝试多次后，Waiter 可能会失败。有关如何使用服务员的示例，请参阅代码示例存储库中的Amazon DynamoDB 加密客户端的 create_table 方法。AWS

调用函数

要调用 Waiter，请在服务客户端上调用 `wait_until`。在以下示例中，Waiter 一直等到实例 `i-12345678` 运行后才继续。

```
ec2 = Aws::EC2::Client.new

begin
  ec2.wait_until(:instance_running, instance_ids:['i-12345678'])
  puts "instance running"
rescue Aws::Waiters::Errors::WaiterFailed => error
  puts "failed waiting for instance running: #{error.message}"
end
```

第一个参数是 Waiter 名称，这特定于服务客户端并指示正在等待哪个操作。第二个参数是传递给 Waiter 所调用的客户端方法的参数的哈希，这因 Waiter 名称而异。

有关可等待的操作和为每个操作调用的客户端方法的列表，请参阅您所使用的客户端的 `waiter_names` 和 `wait_until` 域文档。

等待函数

Waiter 可能会失败并出现以下任何异常。

Aws:: Waiters:: 错误:: FailureStateError

在等待时遇到失败状态。

Aws:: Waiters:: 错误:: NoSuchWaiterError

没有为所使用的客户端定义指定的 Waiter 名称。

Aws:: Waiters:: 错误:: TooManyAttemptsError

尝试次数超过了 Waiter 的 `max_attempts` 值。

Aws:: Waiters:: 错误:: UnexpectedError

在等待时出现意外错误。

Aws:: Waiters:: 错误:: WaiterFailed

在等待时超出了等待状态之一或出现了其他故障。

所有这些错误（除外`NoSuchWaiterError`）都是基于. `WaiterFailed` 要捕获 Waiter 中的错误，请使用 `WaiterFailed`，如以下示例所示。

```
rescue Aws::Waiters::Errors::WaiterFailed => error
  puts "failed waiting for instance running: #{error.message}"
end
```

配置函数

每个 Waiter 都具有默认轮询间隔和它在将控制权返还给您的程序之前将进行尝试的最大次数。要设置这些值，请在您的 `max_attempts` 调用中使用 `delay:` 和 `wait_until` 参数。以下示例最多等待 25 秒，每隔 5 秒轮询一次。

```
# Poll for ~25 seconds
client.wait_until(...) do |w|
  w.max_attempts = 5
  w.delay = 5
end
```

要禁用等待失败，请将其中任一个参数的值设置为 `nil`。

延长函数

要修改 Waiter 的行为，您可以注册在每个轮询尝试之前和等待之前触发的回调。

以下示例通过使每次尝试后的等待时间长度翻倍，在 Waiter 中实现指数回退。

```
ec2 = Aws::EC2::Client.new

ec2.wait_until(:instance_running, instance_ids:['i-12345678']) do |w|
  w.interval = 0 # disable normal sleep
  w.before_wait do |n, resp|
    sleep(n ** 2)
  end
end
```

以下示例禁用了最大尝试次数，而在失败前等待 1 小时（3600 秒）。

```
started_at = Time.now
client.wait_until(...) do |w|
  # Disable max attempts
  w.max_attempts = nil

  # Poll for one hour, instead of a number of attempts
  w.before_wait do |attempts, response|
    throw :failure if Time.now - started_at > 3600
  end
end
```

指定客户端超时时间

默认情况下，AWSSDK for Ruby 最多执行三次重试，两次重试之间有 15 秒，总共最多四次尝试。因此，操作可能需要长达 60 秒才超时。

以下示例在该区域创建了一个 Amazon S3 客户端`us-west-2`，并指定在每次客户端操作的两次重试之间等待五秒钟。因此，Amazon S3 客户端操作最多可能需要 15 秒才能超时。

```
 s3 = Aws::S3::Client.new(
```

```
  region: region,
  retry_limit: 2,
  retry_backoff: lambda { |c| sleep(5) }
)
```

从 AWS SDK for Ruby 版本 1 或 2 迁移到版本 3

本主题的目的是帮助你从AWS适用于 SDK for Ruby 版本 1 或 2 迁移到版本 3。

S ide-by-side 用法

没有必要将适用于 Ruby 的 AWS SDK 的版本 1 或 2 替换为版本 3。您可以在同一应用程序中一起使用它们。有关更多信息，请参阅此博客文章。

下面是一个快速示例。

```
require 'aws-sdk-v1' # version 1
require 'aws-sdk'    # version 2
require 'aws-sdk-s3' # version 3

s3 = AWS::S3::Client.new # version 1
s3 = Aws::S3::Client.new # version 2 or 3
```

您无需重写现有的工作版本 1 或 2 代码即可开始使用版本 3 开发工具包。有效的迁移策略是仅针对版本 3 开发工具包编写新代码。

一般差异

版本 3 在一个重要方面与版本 2 不同。

- 每项服务都作为单独的 Gem 提供。

版本 2 在多个重要方面与版本 1 不同。

- 不同的根命名空间 — `Aws` 与`AWS`. 这样就可以side-by-side使用。
- `Aws.config` – 现在是 vanilla Ruby 哈希，而不是方法。
- 严格的构造函数选项 - 在版本 1 开发工具包中构造客户端或资源对象时，会忽略未知构造函数选项。在版本 2 中，未知构造函数选项会触发 `ArgumentError`。例如：

```
# version 1
AWS::S3::Client.new(http_reed_timeout: 10)
# oops, typo'd option is ignored

# version 2
Aws::S3::Client.new(http_reed_timeout: 10)
# => raises ArgumentError
```

客户差异

版本 2 和版本 3 中的客户端类之间没有区别。

在版本 1 和版本 2 之间，客户端类具有最少的外部区别。在构造客户端后，许多服务客户端都将具有兼容的接口。一些重要区别：

- `Aws::S3::Client`-版本 1 的 Amazon S3 客户端类是手动编码的。版本 2 是从服务模型生成的。方法名称和输入在版本 2 中大不相同。
- `Aws::EC2::Client` - 版本 2 对输出列表使用复数名称，版本 1 使用后缀 `_set`。例如：

```
# version 1
resp = AWS::EC2::Client.new.describe_security_groups
resp.security_group_set
#=> [...]

# version 2
resp = Aws::EC2::Client.new.describe_security_groups
resp.security_groups
#=> [...]
```

- `Aws::SWF::Client` – 版本 2 使用结构化响应，而版本 1 使用 vanilla Ruby 哈希。
- 服务类重命名 – 版本 2 对多个服务使用不同的名称：
  - `AWS::SimpleWorkflow` 变成 `Aws::SWF`
  - `AWS::ELB` 变成 `Aws::ElasticLoadBalancing`
  - `AWS::SimpleEmailService` 变成 `Aws::SES`
- 客户机配置选项-版本 1 中的某些配置选项在版本 2 中被重命名。其他选项被删除或被替换。下面是主要的更改：
  - `:use_ssl` 已删除。版本 2 到处使用 SSL。要禁用 SSL，您必须配置使用 `:endpoint` 的 `http://`。
  - `:ssl_ca_file` 现在是 `:ssl_ca_bundle`
  - `:ssl_ca_path` 现在是 `:ssl_ca_directory`
  - 增加了 `:ssl_ca_store`。
  - `:endpoint` 现在必须是完全限定的 HTTP 或 HTTPS URI 而非主机名。
  - 为每个服务删除了 `:*_port` 选项，现在替换为 `:endpoint`。
  - `:user_agent_prefix` 现在是 `:user_agent_suffix`

资源差异

版本 2 和版本 3 中的资源接口之间没有区别。

版本 1 和版本 2 中的资源接口之间存在显著区别。版本 1 是完全手动编码的，而版本 2 资源接口是从模型生成的。版本 2 资源接口明显更一致。一些系统区别包括：

- 单独的资源类 — 在版本 2 中，服务名称是一个模块，而不是一个类。在此模块中，它为资源接口：

```
# version 1
s3 = AWS::S3.new

# version 2
s3 = Aws::S3::Resource.new
```

- 参考资源 – 版本 2 开发工具包将集合和单个资源 getter 分成两个不同的方法：

```
# version 1
s3.buckets['bucket-name'].objects['key'].delete

# version 2
s3.bucket('bucket-name').object('key').delete
```

- 批量操作-在版本 1 中，所有Batch 操作都是手动编码的实用程序。在版本 2 中，许多批量操作是通过 API 自动生成的批处理操作。版本 2 批处理接口与版本 1 大不相同。

AWS 服务在 AWS SDK for Ruby 中使用

以下部分包含讨论和示例，向您展示如何使用 AWS SDK for Ruby AWS 服务。

如果你不熟悉 AWS SDK for Ruby，你可能需要先通读开始使用 (p. 2)这个主题。

- 带有指导的代码示例 (p. 20)— 为多个示例提供指导性示例AWS 服务。
- 代码示例 (p. 193)— 提供可用服务示例的完整列表（但除了代码之外没有其他指导）。

所有这些示例的源代码都可以在AWS代码示例存储库中下载GitHub。要向 AWS 文档团队提请考虑生成新的代码示例，请创建新的请求。该团队正在寻求生成涵盖更多应用场景和使用情形的代码示例，而不仅仅是涵盖个别 API 调用的简单代码片段。有关说明，请参阅自述文件中的 “提出新代码示例” 部分。GitHub

包含适用于 Ruby 的 AWS SDK 指南的代码示例

本节提供了一些示例，您可以使用适用于 Ruby 的 AWS SDK 进行访问AWS 服务。

在上的代码示例存储库中查找这些示例和其他示例的源AWS代码GitHub。

主题

- CloudTrail示例：使用AWS适用于 SDK for Ruby (p. 20)
- 使用适用于 Ruby 的 AWS SDK 的亚马逊CloudWatch示例 (p. 25)
- CodeBuild示例：使用AWS适用于 SDK for Ruby (p. 47)
- 使用适用于 AWS SDK for Ruby Amazon EC2 示例 (p. 49)
- AWS Elastic Beanstalk示例：使用AWS适用于 SDK for Ruby (p. 85)
- AWS Identity and Access Management(IAM) 使用适用于 Ruby 的AWS开发工具包的示例 (p. 88)
- AWS Key Management Service示例：使用AWS适用于 SDK for Ruby (p. 111)
- AWS Lambda示例：使用AWS适用于 SDK for Ruby (p. 113)
- 使用适用于 Ruby 的 AWS SDK 的Amazon Polly 示例 (p. 117)
- 亚马逊 RDS 示例使用适用于 R AWS SDK for Ruby (p. 120)
- 使用适用于 Ruby 的AWS开发工具包的Amazon S3 示例 (p. 124)
- 使用 Ruby AWS 开发工具包的Amazon SES 示例 (p. 165)
- 使用适用于 Ruby 的AWS软件开发工具包的Amazon SNS (p. 168)
- 使用适用于 S AWS DK for Ruby 的Amazon SQS 示例 (p. 171)
- 亚马逊WorkDocs示例 (p. 190)

CloudTrail示例：使用AWS适用于 SDK for Ruby

CloudTrailAWS 服务是您可以通过获取账户的 AWS API 调用历史记录来监控您在云中的AWS部署。您可以使用以下 AWS SDK for Ruby 代码示例进行访问AWS CloudTrail。有关的更多信息CloudTrail，请参阅 AWS CloudTrail; 文档。

主题

- 列出 CloudTrail 跟踪 (p. 21)
- 创建 CloudTrail 跟踪 (p. 21)
- 列出 CloudTrail 跟踪事件 (p. 23)
- 删除 CloudTrail 跟踪 (p. 24)

列出 CloudTrail 跟踪

此示例使用 describe_trails 方法列出 CloudTrail 跟踪的名称和 CloudTrail 在 us-west-2 区域中将信息存储在其中的存储桶。

选择 Copy 将代码保存在本地。

使用以下代码创建文件 describe_trails.rb。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail'  # v2: require 'aws-sdk'

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

resp = client.describe_trails({})

puts
puts "Found #{resp.trail_list.count} trail(s) in us-west-2:"
puts

resp.trail_list.each do |trail|
  puts 'Name:           ' + trail.name
  puts 'S3 bucket name: ' + trail.s3_bucket_name
  puts
end
```

请参阅上的完整示例GitHub。

创建 CloudTrail 跟踪

此示例使用 create_trail 方法在 us-west-2 区域中创建 CloudTrail 跟踪。它需要两个输入：跟踪的名称和 CloudTrail 将信息存储在其中的存储桶的名称。如果存储桶没有适当的策略，请包括 -p 标志以向存储桶附加正确的策略。

选择 Copy 将代码保存在本地。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
```

```
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail'  # v2: require 'aws-sdk'
require 'aws-sdk-s3'
require 'aws-sdk-sts'

# Attach IAM policy to bucket
def add_policy(bucket)
  # Get account ID using STS
  sts_client = Aws::STS::Client.new(region: 'us-west-2')
  resp = sts_client.get_caller_identity({})
  account_id = resp.account

  # Attach policy to S3 bucket
  s3_client = Aws::S3::Client.new(region: 'us-west-2')

  begin
    policy = {
      'Version' => '2012-10-17',
      'Statement' => [
        {
          'Sid' => 'AWSCloudTrailAclCheck20150319',
          'Effect' => 'Allow',
          'Principal' => {
            'Service' => 'cloudtrail.amazonaws.com',
          },
          'Action' => 's3:GetBucketAcl',
          'Resource' => 'arn:aws:s3:::' + bucket,
        },
        {
          'Sid' => 'AWSCloudTrailWrite20150319',
          'Effect' => 'Allow',
          'Principal' => {
            'Service' => 'cloudtrail.amazonaws.com',
          },
          'Action' => 's3:PutObject',
          'Resource' => 'arn:aws:s3:::' + bucket + '/AWSLogs/' + account_id + '/*',
          'Condition' => {
            'StringEquals' => {
              's3:x-amz-acl' => 'bucket-owner-full-control',
            },
          },
        },
      ]
    }.to_json

    s3_client.put_bucket_policy(
      bucket: bucket,
      policy: policy
    )

    puts 'Successfully added policy to bucket ' + bucket
  rescue StandardError => err
    puts 'Got error trying to add policy to bucket ' + bucket + ':'
    puts err
    exit 1
  end
end
```

```
# main
name = ''
bucket = ''
attach_policy = false

i = 0

while i < ARGV.length
  case ARGV[i]
    when '-b'
      i += 1
      bucket = ARGV[i]

    when '-p'
      attach_policy = true

    else
      name = ARGV[i]
  end

  i += 1
end

if name == '' || bucket == ''
  puts 'You must supply a trail name and bucket name'
  puts USAGE
  exit 1
end

if attach_policy
  add_policy(bucket)
end

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

begin
  client.create_trail({
    name: name, # required
    s3_bucket_name: bucket, # required
  })

  puts 'Successfully created CloudTrail ' + name + ' in us-west-2'
rescue StandardError => err
  puts 'Got error trying to create trail ' + name + ':'
  puts err
  exit 1
end
```

请参阅上的完整示例GitHub。

列出 CloudTrail 跟踪事件

此示例使用 lookup_events 方法来列出 us-west-2 区域中的 CloudTrail 跟踪事件。

选择 Copy 将代码保存在本地。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail'  # v2: require 'aws-sdk'

def show_event(event)
  puts 'Event name:   ' + event.event_name
  puts 'Event ID:     ' + event.event_id
  puts "Event time:   #{event.event_time}"
  puts 'User name:    ' + event.username

  puts 'Resources:'

  event.resources.each do |r|
    puts '  Name:       ' + r.resource_name
    puts '  Type:       ' + r.resource_type
    puts ''
  end
end

# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

resp = client.lookup_events()

puts
puts "Found #{resp.events.count} events in us-west-2:"
puts

resp.events.each do |e|
  show_event(e)
end
```

请参阅上的完整示例GitHub。

删除 CloudTrail 跟踪

此示例使用 delete_trail 方法删除 us-west-2 区域中的 CloudTrail 跟踪。它需要一个输入，即跟踪的名称。

选择 Copy 将代码保存在本地。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-cloudtrail'  # v2: require 'aws-sdk'

if ARGV.length != 1
  puts 'You must supply the name of the trail to delete'
  exit 1
end

name = ARGV[0]
```

```
# Create client in us-west-2
client = Aws::CloudTrail::Client.new(region: 'us-west-2')

begin
  client.delete_trail({
    name: name, # required
  })

  puts 'Successfully deleted CloudTrail ' + name + ' in us-west-2'
rescue StandardError => err
  puts 'Got error trying to delete trail ' + name + ':'
  puts err
  exit 1
end
```

请参阅上的完整示例GitHub。

使用适用于 Ruby 的 AWS SDK 的亚马逊CloudWatch示例

Amazon CloudWatch (CloudWatch) 是针对AWS云资源和您运行的应用程序的监控服务AWS。您可以使用以下示例使用适用于 Ruby 的 AWS SDK 进行访问CloudWatch。有关的更多信息CloudWatch，请参阅亚马逊CloudWatch文档。

主题

- 获取有关亚马逊CloudWatch警报的信息 (p. 25)
- 创建亚马逊CloudWatch警报 (p. 27)
- 启用和禁用亚马逊CloudWatch警报操作 (p. 29)
- 获取有关亚马逊自定义指标的信息 CloudWatch (p. 32)
- 向亚马逊活动发送CloudWatch活动 (p. 34)

获取有关亚马逊CloudWatch警报的信息

以下代码示例显示了有关亚马逊中可用指标警报的信息CloudWatch。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-cloudwatch'

# Displays information about available metric alarms in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts '-' * 16
      puts 'Name:           ' + alarm.alarm_name
      puts 'State value:    ' + alarm.state_value
      puts 'State reason:   ' + alarm.state_reason
      puts 'Metric:         ' + alarm.metric_name
```

```
      puts 'Namespace:      ' + alarm.namespace
      puts 'Statistic:      ' + alarm.statistic
      puts 'Period:         ' + alarm.period.to_s
      puts 'Unit:           ' + alarm.unit.to_s
      puts 'Eval. periods:  ' + alarm.evaluation_periods.to_s
      puts 'Threshold:      ' + alarm.threshold.to_s
      puts 'Comp. operator: ' + alarm.comparison_operator

      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts 'OK actions:'
        alarm.ok_actions.each do |a|
          puts '  ' + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts 'Alarm actions:'
        alarm.alarm_actions.each do |a|
          puts '  ' + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts 'Insufficient data actions:'
        alarm.insufficient_data_actions.each do |a|
          puts '  ' + a
        end
      end

      puts 'Dimensions:'
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts '  Name: ' + d.name + ', Value: ' + d.value
        end
      else
        puts '  None for this alarm.'
      end
    end
  else
    puts 'No alarms found.'
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Full example call:
def run_me
  region = ''

  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby cw-ruby-example-show-alarms.rb REGION'
    puts 'Example: ruby cw-ruby-example-show-alarms.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts 'Available alarms:'
  describe_metric_alarms(cloudwatch_client)
```

```
end

run_me if $PROGRAM_NAME == __FILE__
```

创建亚马逊CloudWatch警报

以下代码示例创建新CloudWatch警报（或更新现有警报，如果指定名称的警报已经存在）。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-cloudwatch'

# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
```

```
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end

# Full example call:
def run_me
  alarm_name = 'ObjectsInBucket'
  alarm_description = 'Objects exist in this bucket for more than 1 day.'
  metric_name = 'NumberOfObjects'
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic']
  namespace = 'AWS/S3'
  statistic = 'Average'
  dimensions = [
    {
      name: 'BucketName',
      value: 'doc-example-bucket'
    },
    {
      name: 'StorageType',
      value: 'AllStorageTypes'
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = 'Count'
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = 'GreaterThanThreshold' # More than one object.
  region = 'us-east-1'

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
```

```
    namespace,
    statistic,
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

启用和禁用亚马逊CloudWatch警报操作

以下代码示例：

1. 创建并启用新CloudWatch警报（或更新现有警报，如果指定名称的警报已经存在）。
2. 禁用新的或现有的警报。要再次启用警报，请致电enable_alarm_actions。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following code example shows how to:
# 1. Create or update an Amazon CloudWatch alarm.
# 2. Disable all actions for an alarm.

require 'aws-sdk-cloudwatch'

# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
```

```
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end

# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
```

```
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Full example call:
def run_me
  alarm_name = 'ObjectsInBucket'
  alarm_description = 'Objects exist in this bucket for more than 1 day.'
  metric_name = 'NumberOfObjects'
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic']
  namespace = 'AWS/S3'
  statistic = 'Average'
  dimensions = [
    {
      name: 'BucketName',
      value: 'doc-example-bucket'
    },
    {
      name: 'StorageType',
      value: 'AllStorageTypes'
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = 'Count'
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = 'GreaterThanThreshold' # More than one object.
  region = 'us-east-1'

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

获取有关亚马逊自定义指标的信息 CloudWatch

以下代码示例：

1. 向中的CloudWatch自定义指标添加数据点。
2. 显示中指标命名空间的可用指标列表CloudWatch。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following example shows how to:
# 1. Add a datapoint to a metric in Amazon CloudWatch.
# 2. List available metrics for a metric namespace in Amazon CloudWatch.

require 'aws-sdk-cloudwatch'

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
```

```
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end

# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts '    Dimensions:'
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts 'No dimensions found.'
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      'Note that it could take up to 15 minutes for recently-added metrics ' \
      'to become available.'
  end
end

# Full example call:
def run_me
  metric_namespace = 'SITE/TRAFFIC'
  region = 'us-east-1'

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts 'Continuing...' unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    'UniqueVisitors',
    'SiteName',
    'example.com',
    5_885.0,
    'Count'
  )

  puts 'Continuing...' unless datapoint_added_to_metric?(
    cloudwatch_client,
```

```
    metric_namespace,
    'UniqueVisits',
    'SiteName',
    'example.com',
    8_628.0,
    'Count'
  )

  puts 'Continuing...' unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    'PageViews',
    'PageURL',
    'example.html',
    18_057.0,
    'Count'
  )

  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

向亚马逊活动发送CloudWatch活动

以下代码示例显示如何在 Amazon Ev CloudWatch ents 中创建和触发规则。每当亚马逊弹性计算云 (Amazon EC2) 中的可用实例更改为运行状态时，该规则都会向亚马逊简单通知服务 (Amazon SNS) 中的指定主题发送通知。此外，相关事件信息将记录到CloudWatch事件中的日志组中。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# The following code example shows how to create and trigger a rule in
# Amazon CloudWatch Events. This rule sends a notification to the specified
# topic in Amazon Simple Notification Service (Amazon SNS) whenever an
# available instance in Amazon Elastic Compute Cloud (Amazon EC2) changes
# to a running state. Also, related event information is logged to a log group
# in Amazon CloudWatch Logs.
#
# This code example works with the following AWS resources through the
# following functions:
#
# - A rule in Amazon CloudWatch Events. See the rule_exists?, rule_found?,
#   create_rule, and display_rule_activity functions.
# - A role in AWS Identity and Access Management (IAM) to allow the rule
#   to work with Amazon CloudWatch Events. See role_exists?, role_found?,
#   and create_role.
# - An Amazon EC2 instance, which triggers the rule whenever it is restarted.
#   See instance_restarted?.
# - A topic and topic subscription in Amazon SNS for the rule to send event
#   notifications to. See topic_exists?, topic_found?, and create_topic.
# - A log group in Amazon CloudWatch Logs to capture related event information.
#   See log_group_exists?, log_group_created?, log_event, and display_log_data.
#
# This code example requires the following AWS resources to exist in advance:
#
# - An Amazon EC2 instance to restart, which triggers the rule.
#
# The run_me function toward the end of this code example calls the
# preceding functions in the correct order.

require 'aws-sdk-sns'
require 'aws-sdk-iam'
```

```
require 'aws-sdk-cloudwatchevents'
require 'aws-sdk-ec2'
require 'aws-sdk-cloudwatch'
require 'aws-sdk-cloudwatchlogs'
require 'securerandom'

# Checks whether the specified Amazon Simple Notification Service
# (Amazon SNS) topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The Amazon Resource Name (ARN) of the
#   topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end
def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end

# Checks whether the specified topic exists among those available to the
# caller in Amazon Simple Notification Service (Amazon SNS).
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The Amazon Resource Name (ARN) of the
#   topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts 'Topic found.'
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts 'Topic found.'
          return true
        end
      end
    end
  end
  puts 'Topic not found.'
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

```
# Creates a topic in Amazon Simple Notification Service (Amazon SNS)
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The Amazon Resource Name (ARN) of the topic that
#   was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
    protocol: 'email',
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts 'Subscription created with ARN ' \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \
    'and confirm the subscription to start receiving notification emails.'
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return 'Error'
end

# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The Amazon Resource Name (ARN) of the
#   role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end

# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The Amazon Resource Name (ARN) of the
#   role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
```

```
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts 'Role found.'
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts 'Role found.'
          return true
        end
      end
    end
  end
  puts 'Role not found.'
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end

# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon CloudWatch Events to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The Amazon Resource Name (ARN) of the role that
#   was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': '2012-10-17',
      'Statement': [
        {
          'Sid': '',
          'Effect': 'Allow',
          'Principal': {
            'Service': 'events.amazonaws.com'
          },
          'Action': 'sts:AssumeRole'
        }
      ]
    }.to_json,
    path: '/',
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts 'Adding access policy to role...'
  iam_client.put_role_policy(
```

```
    policy_document: {
      'Version': '2012-10-17',
      'Statement': [
        {
          'Sid': 'CloudWatchEventsFullAccess',
          'Effect': 'Allow',
          'Resource': '*',
          'Action': 'events:*'
        },
        {
          'Sid': 'IAMPassRoleForCloudWatchEvents',
          'Effect': 'Allow',
          'Resource': 'arn:aws:iam::*:role/AWS_Events_Invoke_Targets',
          'Action': 'iam:PassRole'
        }
      ]
    }.to_json,
    policy_name: 'CloudWatchEventsPolicy',
    role_name: role_name
  )
  puts 'Access policy added to role.'
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts 'If the role was created, you must add the access policy ' \
    'to the role yourself, or delete the role yourself and try again.'
  return 'Error'
end

# Checks whether the specified AWS CloudWatch Events rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end

# Checks whether the specified rule exists among those available to the
# caller in AWS CloudWatch Events.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized AWS CloudWatch Events client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
```

```
      puts 'Rule found.'
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.rules.count.positive?
        if rule_found?(response.rules, rule_name)
          puts 'Rule found.'
          return true
        end
      end
    end
  end
  puts 'Rule not found.'
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end

# Creates a rule in AWS CloudWatch Events.
# This rule is triggered whenever an available instance in
# Amazon Elastic Compute Cloud (Amazon EC2) changes to the specified state.
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon Simple Notification Service (Amazon SNS).
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized AWS CloudWatch Events client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon Elastic Compute Cloud (Amazon EC2) must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
```

```
      'source': [
        'aws.ec2'
      ],
      'detail-type': [
        'EC2 Instance State-change Notification'
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: 'ENABLED',
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
      put_targets_response.failed_entry_count > 0
    puts 'Error(s) adding target to rule:'
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts 'If the rule was created, you must add the target ' \
    'to the rule yourself, or delete the rule yourself and try again.'
  return false
end

# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts 'Log group found.'
        return true
      end
```

```
    end
  end
  puts 'Log group not found.'
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
end

# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts 'Log group created.'
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end

# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
```

```
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts 'Message logged.'
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end

# Restarts an Amazon Elastic Compute Cloud (Amazon EC2) instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ''

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    'This might take a few minutes...'
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts 'Instance stopped.'
  sequence_token = log_event(
    cloudwatchlogs_client,
```

```
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )

  puts 'Attempting to restart the instance. This might take a few minutes...'
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts 'Instance restarted.'
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts 'Error creating log stream or stopping or restarting the instance: ' \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end

# Displays information about activity for a rule in Amazon CloudWatch Events.
#
# Prerequisites:
#
# - A rule in Amazon CloudWatch Events.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts 'Attempting to display rule activity...'
  response = cloudwatch_client.get_metric_statistics(
    namespace: 'AWS/Events',
    metric_name: 'Invocations',
    dimensions: [
```

```
      {
        name: 'RuleName',
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ['Sum'],
    unit: 'Count'
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      'specified time period.'
  end
rescue StandardError => e
  puts "Error getting information about event rule activity: #{e.message}"
end

# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts 'Attempting to display log stream data for the log group ' \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: 'LastEventTime',
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
      describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts '-' * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
          get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
        end
      else
        puts 'No log messages for this log stream.'
      end
    end
```

```
  end
rescue StandardError => e
  puts 'Error getting information about the log streams or their messages: ' \
    "#{e.message}"
end

# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon CloudWatch Events rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts '-' * 10
  puts 'Some of the following AWS resources might still exist in your account.'
  puts 'If you no longer want to use this code example, then to clean up'
  puts 'your AWS account and avoid unexpected costs, you might want to'
  puts 'manually delete any of the following resources if they exist:'
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
  puts "- The Amazon CloudWatch Events rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Full example call:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = 'aws-doc-sdk-examples-topic'
  email_address = 'mary@example.com'
  # Properties for the IAM role.
  role_name = 'aws-doc-sdk-examples-cloudwatch-events-rule-role'
  # Properties for the Amazon CloudWatch Events rule.
  rule_name = 'aws-doc-sdk-examples-ec2-state-change'
  rule_description = 'Triggers when any available EC2 instance starts.'
  instance_state = 'running'
  target_id = 'sns-topic'
  # Properties for the Amazon EC2 instance.
  instance_id = 'i-033c48ef067af3dEX'
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                              # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = 'aws-doc-sdk-examples-cloudwatch-log'
  # AWS service clients for this code example.
  region = 'us-east-1'
  sts_client = Aws::STS::Client.new(region: region)
  sns_client = Aws::SNS::Client.new(region: region)
  iam_client = Aws::IAM::Client.new(region: region)
  cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
  ec2_client = Aws::EC2::Client.new(region: region)
  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
```

```
cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

# Get the caller's account ID for use in forming
# Amazon Resource Names (ARNs) that this code relies on later.
account_id = sts_client.get_caller_identity.account

# If the Amazon SNS topic doesn't exist, create it.
topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
unless topic_exists?(sns_client, topic_arn)
  topic_arn = create_topic(sns_client, topic_name, email_address)
  if topic_arn == 'Error'
    puts 'Could not create the Amazon SNS topic correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam::#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == 'Error'
    puts 'Could not create the IAM role correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Events rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts 'Could not create the Amazon CloudWatch Events rule correctly. ' \
      'Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts 'Could not create the Amazon CloudWatch Logs log group ' \
    'correctly. Program stopped.'
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# Restart the Amazon EC2 instance, which triggers the rule.
unless instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
```

```
    log_group_name
  )
    puts 'Could not restart the instance to trigger the rule. ' \
      'Continuing anyway to show information about the rule and logs...'
  end

  # Display how many times the rule was triggered over the past 10 minutes.
  display_rule_activity(
    cloudwatch_client,
    rule_name,
    start_time,
    end_time,
    period
  )

  # Display related log data in Amazon CloudWatch Logs.
  display_log_data(cloudwatchlogs_client, log_group_name)

  # Reminder the caller to clean up any AWS resources that are used
  # by this code example and are no longer needed.
  manual_cleanup_notice(
    topic_name, role_name, rule_name, log_group_name, instance_id
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

CodeBuild示例：使用AWS适用于 SDK for Ruby

CodeBuild 是一项完全托管的生成服务，可编译源代码、运行测试以及生成可供部署的软件包。您可以使用以下 AWS SDK for Ruby 代码示例进行访问AWS CodeBuild。有关 CodeBuild 的详细信息，请参阅 AWS CodeBuild 文档。

主题

- 获取有关所有 AWS CodeBuild 项目的信息 (p. 47)
- 生成 AWS CodeBuild 项目 (p. 48)
- 列出 AWS CodeBuild 项目生成 (p. 48)

获取有关所有 AWS CodeBuild 项目的信息

以下示例列出了最多 100 个 AWS CodeBuild 项目的名称。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild'  # v2: require 'aws-sdk'

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

resp = client.list_projects({
  sort_by: 'NAME', # accepts NAME, CREATED_TIME, LAST_MODIFIED_TIME
```

```
  sort_order: 'ASCENDING' # accepts ASCENDING, DESCENDING
})

resp.projects.each { |p| puts p }

puts
```

选择 Copy 将代码保存在本地。请参阅上的完整示例GitHub。

生成 AWS CodeBuild 项目

以下示例生成在命令行上指定的 AWS CodeBuild 项目。如果没有提供命令行参数，它会发出错误并退出。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild'  # v2: require 'aws-sdk'

project_name = ''

if ARGV.length != 1
  puts 'You must supply the name of the project to build'
  exit 1
else
  project_name = ARGV[0]
end

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

begin
  client.start_build(project_name: project_name)
  puts 'Building project ' + project_name
rescue StandardError => ex
  puts 'Error building project: ' + ex.message
end
```

选择 Copy 将代码保存在本地。请参阅上的完整示例GitHub。

列出 AWS CodeBuild 项目生成

以下示例显示了有关您的 AWS CodeBuild 项目生成的信息。此类信息包括项目的名称、生成的开始时间以及生成的每个阶段所需的时间（以秒为单位）。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
```

```
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-codebuild'  # v2: require 'aws-sdk'

client = Aws::CodeBuild::Client.new(region: 'us-west-2')

build_list = client.list_builds({sort_order: 'ASCENDING', })

builds = client.batch_get_builds({ids: build_list.ids})

builds.builds.each do |build|
  puts 'Project:    ' + build.project_name
  puts 'Phase:      ' + build.current_phase
  puts 'Status:     ' + build.build_status
end
```

选择 Copy 将代码保存在本地。请参阅上的完整示例GitHub。

使用适用于 AWS SDK for Ruby Amazon EC2 示例

Amazon Elastic Compute Compute Cloud (Amazon EC2) 是一项 Web 服务，可提供可调整的计算容量，实际上是Amazon 数据中心里的服务器，用于构建和托管软件系统。您可以使用以下示例使用适用于 Ruby 的 AWS软件开发工具包访问 Amazon EC2。有关 Amazon EC2 的更多信息，请参阅 Amazon EC2 文档。

主题

- 创建 Amazon EC2 VPC (p. 49)
- 在 Amazon EC2 中创建Internet Gateway 并将其连接到 VPC (p. 51)
- 为 Amazon EC2 创建公有子网 (p. 52)
- 创建 Amazon EC2 路由表并将其与子网关联 (p. 54)
- 在Amazon EC2 中使用弹性 IP 地址 (p. 56)
- 创建 Amazon EC2 安全组 (p. 60)
- 使用 Amazon EC2 安全组 (p. 62)
- 在 Amazon EC2 中使用密钥对 (p. 67)
- 获取有关所有 Amazon EC2 实例的信息 (p. 70)
- 获取有关具有特定标签值的所有 Amazon EC2 实例的信息 (p. 70)
- 获取有关特定 Amazon EC2 实例的信息 (p. 72)
- 创建 Amazon EC2 实例 (p. 73)
- 停止一个 Amazon EC2 实例 (p. 75)
- 启动 Amazon EC2 实例 (p. 76)
- 重新启动 Amazon EC2 实例 (p. 78)
- 管理 Amazon EC2 实例 (p. 79)
- 终止 Amazon EC2 实例 (p. 82)
- 获取有关 Amazon EC2 区域和可用区的信息 (p. 83)

创建 Amazon EC2 VPC

以下代码示例在 Amazon Vual Private Cloud (Amazon VPC) 中创建一个虚拟私有云 (VPC)，然后为 V3 私有云 (VPC)，

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0
```

```
require 'aws-sdk-ec2'

# Creates a virtual private cloud (VPC) in
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param tag_key [String] The key portion of the tag for the VPC.
# @param tag_value [String] The value portion of the tag for the VPC.
# @return [Boolean] true if the VPC was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless vpc_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     '10.0.0.0/24',
#     'my-key',
#     'my-value'
#   )
def vpc_created_and_tagged?(
  ec2_resource,
  cidr_block,
  tag_key,
  tag_value
)
  vpc = ec2_resource.create_vpc(cidr_block: cidr_block)

  # Create a public DNS by enabling DNS support and DNS hostnames.
  vpc.modify_attribute(enable_dns_support: { value: true })
  vpc.modify_attribute(enable_dns_hostnames: { value: true })

  vpc.create_tags(tags: [{ key: tag_key, value: tag_value }])

  puts "Created VPC with ID '#{vpc.id}' and tagged with key " \
    "'#{tag_key}' and value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "#{e.message}"
  return false
end

# Full example call:
def run_me
  cidr_block = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-create-vpc.rb ' \
      'CIDR_BLOCK TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-create-vpc.rb ' \
      '10.0.0.0/24 my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    cidr_block = '10.0.0.0/24'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    cidr_block = ARGV[0]
    tag_key = ARGV[1]
```

```
    tag_value = ARGV[2]
    region = ARGV[3]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if vpc_created_and_tagged?(
    ec2_resource,
    cidr_block,
    tag_key,
    tag_value
  )
    puts 'VPC created and tagged.'
  else
    puts 'VPC not created or not tagged.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon EC2 中创建Internet Gateway 并将其连接到 VPC

以下代码示例创建一个互联网网关，然后将其连接至Amazon Vial VPC ate Cloud (Amazon VPC)。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates an internet gateway and then attaches it to a virtual private cloud
# (VPC) in Amazon Virtual Private Cloud (Amazon VPC).
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC to attach the internet gateway.
# @param tag_key [String] The key of the tag to attach to the internet gateway.
# @param tag_value [String] The value of the tag to attach to the
#   internet gateway.
# @return [Boolean] true if the internet gateway was created and attached;
#   otherwise, false.
# @example
#   exit 1 unless internet_gateway_created_and_attached?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'vpc-6713dfEX'
#   )
def internet_gateway_created_and_attached?(
  ec2_resource,
  vpc_id,
  tag_key,
  tag_value
)
  igw = ec2_resource.create_internet_gateway
  puts "The internet gateway's ID is '#{igw.id}'."
  igw.attach_to_vpc(vpc_id: vpc_id)
  igw.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
```

```
    ]
  )
  return true
rescue StandardError => e
  puts "Error creating or attaching internet gateway: #{e.message}"
  puts 'If the internet gateway was created but not attached, you should ' \
    'clean up by deleting the internet gateway.'
  return false
end

# Full example call:
def run_me
  vpc_id = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-attach-igw-vpc.rb ' \
      'VPC_ID TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-attach-igw-vpc.rb ' \
      'vpc-6713dfEX my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = 'vpc-6713dfEX'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    vpc_id = ARGV[0]
    tag_key = ARGV[1]
    tag_value = ARGV[2]
    region = ARGV[3]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if internet_gateway_created_and_attached?(
    ec2_resource,
    vpc_id,
    tag_key,
    tag_value
  )
    puts "Created and attached internet gateway to VPC '#{vpc_id}'."
  else
    puts "Could not create or attach internet gateway to VPC '#{vpc_id}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

为 Amazon EC2 创建公有子网

以下代码示例在 Amazon Vitual Private Cloud (Amazon Virtual Private Cloud VPC) 中创建一个子网，然后为该子网进行标记。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates a subnet within a virtual private cloud (VPC) in
```

```
# Amazon Virtual Private Cloud (Amazon VPC) and then tags
# the subnet.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the subnet.
# @param cidr_block [String] The IPv4 CIDR block for the subnet.
# @param availability_zone [String] The ID of the Availability Zone
#   for the subnet.
# @param tag_key [String] The key portion of the tag for the subnet.
# @param tag_vlue [String] The value portion of the tag for the subnet.
# @return [Boolean] true if the subnet was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless subnet_created_and_tagged?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'vpc-6713dfEX',
#     '10.0.0.0/24',
#     'us-east-1a',
#     'my-key',
#     'my-value'
#   )
def subnet_created_and_tagged?(
  ec2_resource,
  vpc_id,
  cidr_block,
  availability_zone,
  tag_key,
  tag_value
)
  subnet = ec2_resource.create_subnet(
    vpc_id: vpc_id,
    cidr_block: cidr_block,
    availability_zone: availability_zone
  )
  subnet.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts "Subnet created with ID '#{subnet.id}' in VPC with ID '#{vpc_id}' " \
    "and CIDR block '#{cidr_block}' in availability zone " \
    "'#{availability_zone}' and tagged with key '#{tag_key}' and " \
    "value '#{tag_value}'."
  return true
rescue StandardError => e
  puts "Error creating or tagging subnet: #{e.message}"
  return false
end

# Full example call:
def run_me
  vpc_id = ''
  cidr_block = ''
  availability_zone = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
```

```
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-create-subnet.rb ' \
      'VPC_ID CIDR_BLOCK AVAILABILITY_ZONE TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-create-subnet.rb ' \
      'vpc-6713dfEX 10.0.0.0/24 us-east-1a my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = 'vpc-6713dfEX'
    cidr_block = '10.0.0.0/24'
    availability_zone = 'us-east-1a'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    vpc_id = ARGV[0]
    cidr_block = ARGV[1]
    availability_zone = ARGV[2]
    tag_key = ARGV[3]
    tag_value = ARGV[4]
    region = ARGV[5]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if subnet_created_and_tagged?(
    ec2_resource,
    vpc_id,
    cidr_block,
    availability_zone,
    tag_key,
    tag_value
  )
    puts 'Subnet created and tagged.'
  else
    puts 'Subnet not created or not tagged.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

创建 Amazon EC2 路由表并将其与子网关联

以下代码示例在Amazon Virtual Private Cloud (Amazon VPC) 中创建路由表，然后将该路由表与 Amazon VPC 中的子网相关联。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates a route table in Amazon Virtual Private Cloud (Amazon VPC)
# and then associates the route table with a subnet in Amazon VPC.
#
# Prerequisites:
#
# - A VPC in Amazon VPC.
# - A subnet in that VPC.
# - A gateway attached to that subnet.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon Elastic Compute Cloud (Amazon EC2) resource object.
# @param vpc_id [String] The ID of the VPC for the route table.
```

```
# @param subnet_id [String] The ID of the subnet for the route table.
# @param gateway_id [String] The ID of the gateway for the route.
# @param destination_cidr_block [String] The destination CIDR block
#   for the route.
# @param tag_key [String] The key portion of the tag for the route table.
# @param tag_value [String] The value portion of the tag for the route table.
# @return [Boolean] true if the route table was created and associated;
#   otherwise, false.
# @example
#   exit 1 unless route_table_created_and_associated?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'vpc-0b6f769731EXAMPLE',
#     'subnet-03d9303b57EXAMPLE',
#     'igw-06ca90c011EXAMPLE',
#     '0.0.0.0/0',
#     'my-key',
#     'my-value'
#   )
def route_table_created_and_associated?(
  ec2_resource,
  vpc_id,
  subnet_id,
  gateway_id,
  destination_cidr_block,
  tag_key,
  tag_value
)
  route_table = ec2_resource.create_route_table(vpc_id: vpc_id)
  puts "Created route table with ID '#{route_table.id}'."
  route_table.create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts 'Added tags to route table.'
  route_table.create_route(
    destination_cidr_block: destination_cidr_block,
    gateway_id: gateway_id
  )
  puts 'Created route with destination CIDR block ' \
    "'#{destination_cidr_block}' and associated with gateway " \
    "with ID '#{gateway_id}'."
  route_table.associate_with_subnet(subnet_id: subnet_id)
  puts "Associated route table with subnet with ID '#{subnet_id}'."
  return true
rescue StandardError => e
  puts "Error creating or associating route table: #{e.message}"
  puts 'If the route table was created but not associated, you should ' \
    'clean up by deleting the route table.'
  return false
end

# Full example call:
def run_me
  vpc_id = ''
  subnet_id = ''
  gateway_id = ''
  destination_cidr_block = ''
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
```

```
    puts 'Usage: ruby ec2-ruby-example-create-route-table.rb ' \
      'VPC_ID SUBNET_ID GATEWAY_ID DESTINATION_CIDR_BLOCK ' \
      'TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-create-route-table.rb ' \
      'vpc-0b6f769731EXAMPLE subnet-03d9303b57EXAMPLE igw-06ca90c011EXAMPLE ' \
      '\'0.0.0.0/0\' my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    vpc_id = 'vpc-0b6f769731EXAMPLE'
    subnet_id = 'subnet-03d9303b57EXAMPLE'
    gateway_id = 'igw-06ca90c011EXAMPLE'
    destination_cidr_block = '0.0.0.0/0'
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    vpc_id = ARGV[0]
    subnet_id = ARGV[1]
    gateway_id = ARGV[2]
    destination_cidr_block = ARGV[3]
    tag_key = ARGV[4]
    tag_value = ARGV[5]
    region = ARGV[6]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if route_table_created_and_associated?(
    ec2_resource,
    vpc_id,
    subnet_id,
    gateway_id,
    destination_cidr_block,
    tag_key,
    tag_value
  )
    puts 'Route table created and associated.'
  else
    puts 'Route table not created or not associated.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在Amazon EC2 中使用弹性 IP 地址

以下代码示例：

1. 显示与Amazon Elastic Compute Cloud (Amazon EC2) 实例关联的任何地址的信息。
2. 在Amazon Virte Cloud (Amazon VPC) 中创建弹性 IP 地址。
3. 将该地址与实例关联起来。
4. 再次显示与实例关联的地址的信息。这次，应该会显示新的地址关联。
5. 发布地址。
6. 再次显示与实例关联的地址的信息。这次，不应显示已发布的地址。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0
```

```
# This code example does the following:
# 1. Displays information about any addresses associated with an
#    Amazon Elastic Compute Cloud (Amazon EC2) instance.
# 2. Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
# 3. Associates the address with the instance.
# 4. Displays information again about addresses associated with the instance.
#    This time, the new address association should display.
# 5. Releases the address.
# 6. Displays information again about addresses associated with the instance.
#    This time, the released address should not display.

require 'aws-sdk-ec2'

# Checks whether the specified Amazon Elastic Compute Cloud
# (Amazon EC2) instance exists.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance exists; otherwise, false.
# @example
#   exit 1 unless instance_exists?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_exists?(ec2_client, instance_id)
  ec2_client.describe_instances(instance_ids: [instance_id])
  return true
rescue StandardError
  return false
end

# Creates an Elastic IP address in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @return [String] The allocation ID corresponding to the Elastic IP address.
# @example
#   puts allocate_elastic_ip_address(Aws::EC2::Client.new(region: 'us-east-1'))
def allocate_elastic_ip_address(ec2_client)
  response = ec2_client.allocate_address(domain: 'vpc')
  return response.allocation_id
rescue StandardError => e
  puts "Error allocating Elastic IP address: #{e.message}"
  return 'Error'
end

# Associates an Elastic IP address with an Amazon Elastic Compute Cloud
# (Amazon EC2) instance.
#
# Prerequisites:
#
# - The allocation ID corresponding to the Elastic IP address.
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @param instance_id [String] The ID of the instance.
# @return [String] The assocation ID corresponding to the association of the
#   Elastic IP address to the instance.
# @example
#   puts allocate_elastic_ip_address(
#     Aws::EC2::Client.new(region: 'us-east-1'),
```

```
#     'eipalloc-04452e528a66279EX',
#     'i-033c48ef067af3dEX')
def associate_elastic_ip_address_with_instance(
  ec2_client,
  allocation_id,
  instance_id
)
  response = ec2_client.associate_address(
    allocation_id: allocation_id,
    instance_id: instance_id,
  )
  return response.association_id
rescue StandardError => e
  puts "Error associating Elastic IP address with instance: #{e.message}"
  return 'Error'
end

# Gets information about addresses associated with an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
#   describe_addresses_for_instance(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def describe_addresses_for_instance(ec2_client, instance_id)
  response = ec2_client.describe_addresses(
    filters: [
      {
        name: 'instance-id',
        values: [instance_id]
      }
    ]
  )
  addresses = response.addresses
  if addresses.count.zero?
    puts 'No addresses.'
  else
    addresses.each do |address|
      puts '-' * 20
      puts "Public IP:  #{address.public_ip}"
      puts "Private IP: #{address.private_ip_address}"
    end
  end
rescue StandardError => e
  puts "Error getting address information for instance: #{e.message}"
end

# Releases an Elastic IP address from an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance with an associated Elastic IP address.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param allocation_id [String] The ID of the allocation corresponding to
#   the Elastic IP address.
# @return [Boolean] true if the Elastic IP address was released;
#   otherwise, false.
```

```
# @example
#   exit 1 unless elastic_ip_address_released?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'eipalloc-04452e528a66279EX'
#   )
def elastic_ip_address_released?(ec2_client, allocation_id)
  ec2_client.release_address(allocation_id: allocation_id)
  return true
rescue StandardError => e
  return "Error releasing Elastic IP address: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-elastic-ips.rb ' \
      'INSTANCE_ID REGION'
    puts 'Example: ruby ec2-ruby-example-elastic-ips.rb ' \
      'i-033c48ef067af3dEX us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-033c48ef067af3dEX'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  unless instance_exists?(ec2_client, instance_id)
    puts "Cannot find instance with ID '#{instance_id}'. Stopping program."
    exit 1
  end

  puts "Addresses for instance with ID '#{instance_id}' before allocating " \
    'Elastic IP address:'
  describe_addresses_for_instance(ec2_client, instance_id)

  puts 'Allocating Elastic IP address...'
  allocation_id = allocate_elastic_ip_address(ec2_client)
  if allocation_id.start_with?('Error')
    puts 'Stopping program.'
    exit 1
  else
    puts "Elastic IP address created with allocation ID '#{allocation_id}'."
  end

  puts 'Associating Elastic IP address with instance...'
  association_id = associate_elastic_ip_address_with_instance(
    ec2_client,
    allocation_id,
    instance_id
  )
  if association_id.start_with?('Error')
    puts 'Stopping program. You must associate the Elastic IP address yourself.'
    exit 1
  else
    puts 'Elastic IP address associated with instance with association ID ' \
      "'#{association_id}'."
```

```
  end

  puts 'Addresses for instance after allocating Elastic IP address:'
  describe_addresses_for_instance(ec2_client, instance_id)

  puts 'Releasing the Elastic IP address from the instance...'
  if elastic_ip_address_released?(ec2_client, allocation_id) == false
    puts 'Stopping program. You must release the Elastic IP address yourself.'
    exit 1
  else
    puts 'Address released.'
  end

  puts 'Addresses for instance after releasing Elastic IP address:'
  describe_addresses_for_instance(ec2_client, instance_id)
end

run_me if $PROGRAM_NAME == __FILE__
```

创建 Amazon EC2 安全组

以下代码示例创建 Amazon EC2 安全组，然后向该安全组添加出站规则。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group and
# then adds an outbound rule to that security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_resource [Aws::EC2::Resource] An initialized
#   Amazon EC2 resource object.
# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.
# @param vpc_id [String] The ID of the VPC for the security group.
# @param protocol [String] The network protocol for the outbound rule.
# @param from_port [String] The originating port for the outbound rule.
# @param to_port [String] The destination port for the outbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the outbound rule.
# @return [Boolean] true if the security group was created and the outbound
#   rule was added; otherwise, false.
# @example
#   exit 1 unless security_group_created_with_egress?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX',
#     'tcp',
#     '22',
#     '22',
#     '0.0.0.0/0'
#   )
def security_group_created_with_egress?(
  ec2_resource,
  group_name,
  description,
  vpc_id,
  ip_protocol,
  from_port,
```

```
  to_port,
  cidr_ip_range
)
  security_group = ec2_resource.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.id}' in VPC with ID '#{vpc_id}'."
  security_group.authorize_egress(
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
            cidr_ip: cidr_ip_range
          }
        ]
      }
    ]
  )
  puts "Granted egress to security group '#{group_name}' for protocol " \
    "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
    "with CIDR IP range '#{cidr_ip_range}'."
  return true
rescue StandardError => e
  puts "Error creating security group or granting egress: #{e.message}"
  return false
end

# Full example call:
def run_me
  group_name = ''
  description = ''
  vpc_id = ''
  ip_protocol = ''
  from_port = ''
  to_port = ''
  cidr_ip_range = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-create-security-group.rb ' \
      'GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL FROM_PORT TO_PORT ' \
      'CIDR_IP_RANGE REGION'
    puts 'Example: ruby ec2-ruby-example-create-security-group.rb ' \
      'my-security-group \'This is my security group.\' vpc-6713dfEX ' \
      'tcp 22 22 \'0.0.0.0/0\' us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    group_name = 'my-security-group'
    description = 'This is my security group.'
    vpc_id = 'vpc-6713dfEX'
    ip_protocol = 'tcp'
    from_port = '22'
    to_port = '22'
    cidr_ip_range = '0.0.0.0/0'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    group_name = ARGV[0]
    description = ARGV[1]
```

```
    vpc_id = ARGV[2]
    ip_protocol = ARGV[3]
    from_port = ARGV[4]
    to_port = ARGV[5]
    cidr_ip_range = ARGV[6]
    region = ARGV[7]
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if security_group_created_with_egress?(
    ec2_resource,
    group_name,
    description,
    vpc_id,
    ip_protocol,
    from_port,
    to_port,
    cidr_ip_range
  )
    puts 'Security group created and egress granted.'
  else
    puts 'Security group not created or egress not granted.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 Amazon EC2 安全组

以下示例：

1. 创建 Amazon EC2 安全组。
2. 将入站规则添加到安全组。
3. 显示有关可用安全组的信息。
4. 删除安全组。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
# 2. Adds inbound rules to the security group.
# 3. Displays information about available security groups.
# 4. Deletes the security group.

require 'aws-sdk-ec2'

# Creates an Amazon Elastic Compute Cloud (Amazon EC2) security group.
#
# Prerequisites:
#
# - A VPC in Amazon Virtual Private Cloud (Amazon VPC).
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param group_name [String] A name for the security group.
# @param description [String] A description for the security group.
# @param vpc_id [String] The ID of the VPC for the security group.
# @return [String] The ID of security group that was created.
# @example
```

```
#   puts create_security_group(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-security-group',
#     'This is my security group.',
#     'vpc-6713dfEX'
#   )
def create_security_group(
  ec2_client,
  group_name,
  description,
  vpc_id
)
  security_group = ec2_client.create_security_group(
    group_name: group_name,
    description: description,
    vpc_id: vpc_id
  )
  puts "Created security group '#{group_name}' with ID " \
    "'#{security_group.group_id}' in VPC with ID '#{vpc_id}'."
  return security_group.group_id
rescue StandardError => e
  puts "Error creating security group: #{e.message}"
  return 'Error'
end

# Adds an inbound rule to an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param security_group_id [String] The ID of the security group.
# @param ip_protocol [String] The network protocol for the inbound rule.
# @param from_port [String] The originating port for the inbound rule.
# @param to_port [String] The destination port for the inbound rule.
# @param cidr_ip_range [String] The CIDR IP range for the inbound rule.
# @return
# @example
#   exit 1 unless security_group_ingress_authorized?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'sg-030a858e078f1b9EX',
#     'tcp',
#     '80',
#     '80',
#     '0.0.0.0/0'
#   )
def security_group_ingress_authorized?(
  ec2_client,
  security_group_id,
  ip_protocol,
  from_port,
  to_port,
  cidr_ip_range
)
  ec2_client.authorize_security_group_ingress(
    group_id: security_group_id,
    ip_permissions: [
      {
        ip_protocol: ip_protocol,
        from_port: from_port,
        to_port: to_port,
        ip_ranges: [
          {
            cidr_ip: cidr_ip_range
```

```
          }
        ]
      }
    ]
  )
  puts "Added inbound rule to security group '#{security_group_id}' for protocol " \
    "'#{ip_protocol}' from port '#{from_port}' to port '#{to_port}' " \
    "with CIDR IP range '#{cidr_ip_range}'."
  return true
rescue StandardError => e
  puts "Error adding inbound rule to security group: #{e.message}"
  return false
end

# Displays information about a security group's IP permissions set in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - A security group with inbound rules, outbound rules, or both.
#
# @param p [Aws::EC2::Types::IpPermission] The IP permissions set.
# @example
#   ec2_client = Aws::EC2::Client.new(region: 'us-east-1')
#   response = ec2_client.describe_security_groups
#   unless sg.ip_permissions.empty?
#     describe_security_group_permissions(
#       response.security_groups[0].ip_permissions[0]
#     )
#   end
def describe_security_group_permissions(perm)
  print "  Protocol: #{perm.ip_protocol == '-1' ? 'All' : perm.ip_protocol}"

  unless perm.from_port.nil?
    if perm.from_port == '-1' || perm.from_port == -1
      print ', From: All'
    else
      print ", From: #{perm.from_port}"
    end
  end

  unless perm.to_port.nil?
    if perm.to_port == '-1' || perm.to_port == -1
      print ', To: All'
    else
      print ", To: #{perm.to_port}"
    end
  end

  if perm.key?(:ipv_6_ranges) && perm.ipv_6_ranges.count.positive?
    print ", CIDR IPv6: #{perm.ipv_6_ranges[0].cidr_ipv_6}"
  end

  if perm.key?(:ip_ranges) && perm.ip_ranges.count.positive?
    print ", CIDR IPv4: #{perm.ip_ranges[0].cidr_ip}"
  end

  print "\n"
end

# Displays information about available security groups in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @example
#   describe_security_groups(Aws::EC2::Client.new(region: 'us-east-1'))
```

```
def describe_security_groups(ec2_client)
  response = ec2_client.describe_security_groups

  if response.security_groups.count.positive?
    response.security_groups.each do |sg|
      puts '-' * (sg.group_name.length + 13)
      puts "Name:        #{sg.group_name}"
      puts "Description: #{sg.description}"
      puts "Group ID:    #{sg.group_id}"
      puts "Owner ID:    #{sg.owner_id}"
      puts "VPC ID:      #{sg.vpc_id}"

      if sg.tags.count.positive?
        puts 'Tags:'
        sg.tags.each do |tag|
          puts "  Key: #{tag.key}, Value: #{tag.value}"
        end
      end

      unless sg.ip_permissions.empty?
        puts 'Inbound rules:' if sg.ip_permissions.count.positive?
        sg.ip_permissions.each do |p|
          describe_security_group_permissions(p)
        end
      end

      unless sg.ip_permissions_egress.empty?
        puts 'Outbound rules:' if sg.ip_permissions.count.positive?
        sg.ip_permissions_egress.each do |p|
          describe_security_group_permissions(p)
        end
      end
    end
  else
    puts 'No security groups found.'
  end
rescue StandardError => e
  puts "Error getting information about security groups: #{e.message}"
end

# Deletes an Amazon Elastic Compute Cloud (Amazon EC2)
# security group.
#
# Prerequisites:
#
# - The security group.
#
# @param ec2_client [Aws::EC2::Client] An initialized
#   Amazon EC2 client.
# @param security_group_id [String] The ID of the security group to delete.
# @return [Boolean] true if the security group was deleted; otherwise, false.
# @example
#   exit 1 unless security_group_deleted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'sg-030a858e078f1b9EX'
#   )
def security_group_deleted?(ec2_client, security_group_id)
  ec2_client.delete_security_group(group_id: security_group_id)
  puts "Deleted security group '#{security_group_id}'."
  return true
rescue StandardError => e
  puts "Error deleting security group: #{e.message}"
  return false
end

# Full example call:
```

```
def run_me
  group_name = ''
  description = ''
  vpc_id = ''
  ip_protocol_http = ''
  from_port_http = ''
  to_port_http = ''
  cidr_ip_range_http = ''
  ip_protocol_ssh = ''
  from_port_ssh = ''
  to_port_ssh = ''
  cidr_ip_range_ssh = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-security-group.rb ' \
      'GROUP_NAME DESCRIPTION VPC_ID IP_PROTOCOL_1 FROM_PORT_1 TO_PORT_1 ' \
      'CIDR_IP_RANGE_1 IP_PROTOCOL_2 FROM_PORT_2 TO_PORT_2 ' \
      'CIDR_IP_RANGE_2 REGION'
    puts 'Example: ruby ec2-ruby-example-security-group.rb ' \
      'my-security-group \'This is my security group.\' vpc-6713dfEX ' \
      'tcp 80 80 \'0.0.0.0/0\' tcp 22 22 \'0.0.0.0/0\' us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    group_name = 'my-security-group'
    description = 'This is my security group.'
    vpc_id = 'vpc-6713dfEX'
    ip_protocol_http = 'tcp'
    from_port_http = '80'
    to_port_http = '80'
    cidr_ip_range_http = '0.0.0.0/0'
    ip_protocol_ssh = 'tcp'
    from_port_ssh = '22'
    to_port_ssh = '22'
    cidr_ip_range_ssh = '0.0.0.0/0'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    group_name = ARGV[0]
    description = ARGV[1]
    vpc_id = ARGV[2]
    ip_protocol_http = ARGV[3]
    from_port_http = ARGV[4]
    to_port_http = ARGV[5]
    cidr_ip_range_http = ARGV[6]
    ip_protocol_ssh = ARGV[7]
    from_port_ssh = ARGV[8]
    to_port_ssh = ARGV[9]
    cidr_ip_range_ssh = ARGV[10]
    region = ARGV[11]
  end

  security_group_id = ''
  security_group_exists = false
  ec2_client = Aws::EC2::Client.new(region: region)

  puts 'Attempting to create security group...'
  security_group_id = create_security_group(
    ec2_client,
    group_name,
    description,
    vpc_id
  )
  if security_group_id == 'Error'
    puts 'Could not create security group. Skipping this step.'
```

```
  else
    security_group_exists = true
  end

  if security_group_exists
    puts 'Attempting to add inbound rules to security group...'
    unless security_group_ingress_authorized?(
      ec2_client,
      security_group_id,
      ip_protocol_http,
      from_port_http,
      to_port_http,
      cidr_ip_range_http
    )
      puts 'Could not add inbound HTTP rule to security group. ' \
        'Skipping this step.'
    end

    unless security_group_ingress_authorized?(
      ec2_client,
      security_group_id,
      ip_protocol_ssh,
      from_port_ssh,
      to_port_ssh,
      cidr_ip_range_ssh
    )
      puts 'Could not add inbound SSH rule to security group. ' \
        'Skipping this step.'
    end
  end

  puts "\nInformation about available security groups:"
  describe_security_groups(ec2_client)

  if security_group_exists
    puts "\nAttempting to delete security group..."
    unless security_group_deleted?(ec2_client, security_group_id)
      puts 'Could not delete security group. You must delete it yourself.'
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon EC2 中使用密钥对

以下代码示例：

1. 在 Amazon EC2 中创建key pair。
2. 显示有关可用密钥对的信息。
3. 删除key pair。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
# 2. Displays information about available key pairs.
# 3. Deletes the key pair.

require 'aws-sdk-ec2'
```

```
# Creates a key pair in Amazon Elastic Compute Cloud (Amazon EC2) and
# saves the resulting RSA private key file locally in the calling
# user's home directory.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name for the key pair and private
#   key file.
# @return [Boolean] true if the key pair and private key file were
#   created; otherwise, false.
# @example
#   exit 1 unless key_pair_created?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-key-pair'
#   )
def key_pair_created?(ec2_client, key_pair_name)
  key_pair = ec2_client.create_key_pair(key_name: key_pair_name)
  puts "Created key pair '#{key_pair.key_name}' with fingerprint " \
    "'#{key_pair.key_fingerprint}' and ID '#{key_pair.key_pair_id}'."
  filename = File.join(Dir.home, key_pair_name + '.pem')
  File.open(filename, 'w') { |file| file.write(key_pair.key_material) }
  puts "Private key file saved locally as '#{filename}'."
  return true
rescue Aws::EC2::Errors::InvalidKeyPairDuplicate
  puts "Error creating key pair: a key pair named '#{key_pair_name}' " \
    'already exists.'
  return false
rescue StandardError => e
  puts "Error creating key pair or saving private key file: #{e.message}"
  return false
end

# Displays information about available key pairs in
# Amazon Elastic Compute Cloud (Amazon EC2).
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   describe_key_pairs(Aws::EC2::Client.new(region: 'us-east-1'))
def describe_key_pairs(ec2_client)
  result = ec2_client.describe_key_pairs
  if result.key_pairs.count.zero?
    puts 'No key pairs found.'
  else
    puts 'Key pair names:'
    result.key_pairs.each do |key_pair|
      puts key_pair.key_name
    end
  end
rescue StandardError => e
  puts "Error getting information about key pairs: #{e.message}"
end

# Deletes a key pair in Amazon Elastic Compute Cloud (Amazon EC2).
#
# Prerequisites:
#
# - The key pair to delete.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param key_pair_name [String] The name of the key pair to delete.
# @return [Boolean] true if the key pair was deleted; otherwise, false.
# @example
#   exit 1 unless key_pair_deleted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'my-key-pair'
#   )
def key_pair_deleted?(ec2_client, key_pair_name)
```

```
  ec2_client.delete_key_pair(key_name: key_pair_name)
  return true
rescue StandardError => e
  puts "Error deleting key pair: #{e.message}"
  return false
end

# Full example call:
def run_me
  key_pair_name = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-key-pairs.rb KEY_PAIR_NAME REGION'
    puts 'Example: ruby ec2-ruby-example-key-pairs.rb my-key-pair us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    key_pair_name = 'my-key-pair'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    key_pair_name = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts 'Displaying existing key pair names before creating this key pair...'
  describe_key_pairs(ec2_client)

  puts '-' * 10
  puts 'Creating key pair...'
  unless key_pair_created?(ec2_client, key_pair_name)
    puts 'Stopping program.'
    exit 1
  end

  puts '-' * 10
  puts 'Displaying existing key pair names after creating this key pair...'
  describe_key_pairs(ec2_client)

  puts '-' * 10
  puts 'Deleting key pair...'
  unless key_pair_deleted?(ec2_client, key_pair_name)
    puts 'Stopping program. You must delete the key pair yourself.'
    exit 1
  end
  puts 'Key pair deleted.'

  puts '-' * 10
  puts 'Now that the key pair is deleted, ' \
    'also deleting the related private key pair file...'
  filename = File.join(Dir.home, key_pair_name + '.pem')
  File.delete(filename)
  if File.exist?(filename)
    puts "Could not delete file at '#{filename}'. You must delete it yourself."
  else
    puts 'File deleted.'
  end

  puts '-' * 10
  puts 'Displaying existing key pair names after deleting this key pair...'
  describe_key_pairs(ec2_client)
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

获取有关所有 Amazon EC2 实例的信息

以下代码示例列出了可用 Amazon EC2 实例的 ID 和当前状态。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the IDs and current states of available
# Amazon Elastic Compute Cloud (Amazon EC2) instances.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @example
#   list_instance_ids_states(Aws::EC2::Resource.new(region: 'us-east-1'))
def list_instance_ids_states(ec2_resource)
  response = ec2_resource.instances
  if response.count.zero?
    puts 'No instances found.'
  else
    puts 'Instances -- ID, state:'
    response.each do |instance|
      puts "#{instance.id}, #{instance.state.name}"
    end
  end
rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

#Full example call:
def run_me
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-get-all-instance-info.rb REGION'
    puts 'Example: ruby ec2-ruby-example-get-all-instance-info.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end
  ec2_resource = Aws::EC2::Resource.new(region: region)
  list_instance_ids_states(ec2_resource)
end

run_me if $PROGRAM_NAME == __FILE__
```

获取有关具有特定标签值的所有 Amazon EC2 实例的信息

以下代码示例列出了与指定标签键和值匹配的可用 Amazon EC2 实例的 ID 和当前状态。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the IDs, current states, and tag keys/values of matching
```

```
# available Amazon Elastic Compute Cloud (Amazon EC2) instances.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @param tag_key [String] The key portion of the tag to search on.
# @param tag_value [String] The value portion of the tag to search on.
# @example
#   list_instance_ids_states_by_tag(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'my-key',
#     'my-value'
#   )
def list_instance_ids_states_by_tag(ec2_resource, tag_key, tag_value)
  response = ec2_resource.instances(
    filters: [
      {
        name: "tag:#{tag_key}",
        values: [tag_value]
      }
    ]
  )
  if response.count.zero?
    puts 'No matching instances found.'
  else
    puts 'Matching instances -- ID, state, tag key/value:'
    response.each do |instance|
      print "#{instance.id}, #{instance.state.name}"
      instance.tags.each do |tag|
        print ", #{tag.key}/#{tag.value}"
      end
      print "\n"
    end
  end
rescue StandardError => e
  puts "Error getting information about instances: #{e.message}"
end

#Full example call:
def run_me
  tag_key = ''
  tag_value = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-get-instance-info-by-tag.rb ' \
      'TAG_KEY TAG_VALUE REGION'
    puts 'Example: ruby ec2-ruby-example-get-instance-info-by-tag.rb ' \
      'my-key my-value us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    tag_key = 'my-key'
    tag_value = 'my-value'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    tag_key = ARGV[0]
    tag_value = ARGV[1]
    region = ARGV[2]
  end
  ec2_resource = Aws::EC2::Resource.new(region: region)
  list_instance_ids_states_by_tag(ec2_resource, tag_key, tag_value)
end

run_me if $PROGRAM_NAME == __FILE__
```

获取有关特定 Amazon EC2 实例的信息

以下示例列出指定 Amazon EC2 实例的状态。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Lists the state of an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
#   list_instance_state(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def list_instance_state(ec2_client, instance_id)
  response = ec2_client.describe_instances(
    instance_ids: [instance_id]
  )
  if response.count.zero?
    puts 'No matching instance found.'
  else
    instance = response.reservations[0].instances[0]
    puts "The instance with ID '#{instance_id}' is '#{instance.state.name}'."
  end
rescue StandardError => e
  puts "Error getting information about instance: #{e.message}"
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-list-state-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION'
    puts 'Example: ruby ec2-ruby-example-list-state-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)
  list_instance_state(ec2_client, instance_id)
end

run_me if $PROGRAM_NAME == __FILE__
```

创建 Amazon EC2 实例

以下示例创建并标记 Amazon EC2 实例。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'
require 'base64'

# Creates and tags an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An EC2 key pair.
# - If you want to run any commands on the instance after it starts, a
#   file containing those commands.
#
# @param ec2_resource [Aws::EC2::Resource] An initialized EC2 resource object.
# @param image_id [String] The ID of the target Amazon Machine Image (AMI).
# @param key_pair_name [String] The name of the existing EC2 key pair.
# @param tag_key [String] The key portion of the tag for the instance.
# @param tag_value [String] The value portion of the tag for the instance.
# @param instance_type [String] The ID of the type of instance to create.
#   If not specified, the default value is 't2.micro'.
# @param user_data_file [String] The path to the file containing any commands
#   to run on the instance after it starts. If not specified, the default
#   value is an empty string.
# @return [Boolean] true if the instance was created and tagged;
#   otherwise, false.
# @example
#   exit 1 unless instance_created?(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'ami-0947d2ba12EXAMPLE',
#     'my-key-pair',
#     'my-key',
#     'my-value',
#     't2.micro',
#     'my-user-data.txt'
#   )
def instance_created?(
  ec2_resource,
  image_id,
  key_pair_name,
  tag_key,
  tag_value,
  instance_type = 't2.micro',
  user_data_file = ''
)
  encoded_script = ''

  unless user_data_file == ''
    script = File.read(user_data_file)
    encoded_script = Base64.encode64(script)
  end

  instance = ec2_resource.create_instances(
    image_id: image_id,
    min_count: 1,
    max_count: 1,
    key_name: key_pair_name,
    instance_type: instance_type,
    user_data: encoded_script
  )
```

```
  puts 'Creating instance...'

  # Check whether the new instance is in the "running" state.
  polls = 0
  loop do
    polls += 1
    response = ec2_resource.client.describe_instances(
      instance_ids: [
        instance.first.id
      ]
    )
    # Stop polling after 10 minutes (40 polls * 15 seconds per poll) if not running.
    break if response.reservations[0].instances[0].state.name == 'running' || polls > 40

    sleep(15)
  end

  puts "Instance created with ID '#{instance.first.id}'."

  instance.batch_create_tags(
    tags: [
      {
        key: tag_key,
        value: tag_value
      }
    ]
  )
  puts 'Instance tagged.'

  return true
rescue StandardError => e
  puts "Error creating or tagging instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  image_id = ''
  key_pair_name = ''
  tag_key = ''
  tag_value = ''
  instance_type = ''
  region = ''
  user_data_file = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage: ruby ec2-ruby-example-create-instance.rb ' \
      'IMAGE_ID KEY_PAIR_NAME TAG_KEY TAG_VALUE INSTANCE_TYPE ' \
      'REGION [USER_DATA_FILE]'
    puts 'Example: ruby ec2-ruby-example-create-instance.rb ' \
      'ami-0947d2ba12EXAMPLE my-key-pair my-key my-value t2.micro ' \
      'us-east-1 my-user-data.txt'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    image_id = 'ami-0947d2ba12EXAMPLE'
    key_pair_name = 'my-key-pair'
    tag_key = 'my-key'
    tag_value = 'my-value'
    instance_type = 't2.micro'
    region = 'us-east-1'
    user_data_file = 'my-user-data.txt'
  # Otherwise, use the values as specified at the command prompt.
  else
    image_id = ARGV[0]
```

```
    key_pair_name = ARGV[1]
    tag_key = ARGV[2]
    tag_value = ARGV[3]
    instance_type = ARGV[4]
    region = ARGV[5]
    user_data_file = ARGV[6] if ARGV.count == 7 # If user data file specified.
  end

  ec2_resource = Aws::EC2::Resource.new(region: region)

  if instance_created?(
    ec2_resource,
    image_id,
    key_pair_name,
    tag_key,
    tag_value,
    instance_type,
    user_data_file
  )
    puts 'Created and tagged instance.'
  else
    puts 'Could not create or tag instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

停止一个 Amazon EC2 实例

以下示例尝试停止指定的 Amazon EC2 实例。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to stop an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_stopped?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when 'stopping'
      puts 'The instance is already stopping.'
      return true
    when 'stopped'
      puts 'The instance is already stopped.'
      return true
    when 'terminated'
      puts 'Error stopping instance: ' \
```

```
        'the instance is terminated, so you cannot stop it.'
      return false
    end
  end

  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts 'Instance stopped.'
  return true
rescue StandardError => e
  puts "Error stopping instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-stop-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to stop instance '#{instance_id}' " \
    '(this might take a few minutes)...'
  unless instance_stopped?(ec2_client, instance_id)
    puts 'Could not stop instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

启动 Amazon EC2 实例

以下示例尝试启动指定的 Amazon EC2 实例。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to start an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was started; otherwise, false.
```

```
# @example
#   exit 1 unless instance_started?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_started?(ec2_client, instance_id)
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive?
    state = response.instance_statuses[0].instance_state.name
    case state
    when 'pending'
      puts 'Error starting instance: the instance is pending. Try again later.'
      return false
    when 'running'
      puts 'The instance is already running.'
      return true
    when 'terminated'
      puts 'Error starting instance: ' \
        'the instance is terminated, so you cannot start it.'
      return false
    end
  end

  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts 'Instance started.'
  return true
rescue StandardError => e
  puts "Error starting instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-start-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to start instance '#{instance_id}' " \
    '(this might take a few minutes)...'
  unless instance_started?(ec2_client, instance_id)
    puts 'Could not start instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

重新启动 Amazon EC2 实例

以下示例尝试重启指定的 Amazon EC2 实例。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Reboots an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - An Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @example
#   request_instance_reboot(
#     Aws::EC2::Resource.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def request_instance_reboot(ec2_client, instance_id)
  response = ec2_client.describe_instances(instance_ids: [instance_id])
  if response.count.zero?
    puts 'Error requesting reboot: no matching instance found.'
  else
    instance = response.reservations[0].instances[0]
    if instance.state.name == 'terminated'
      puts 'Error requesting reboot: the instance is already terminated.'
    else
      ec2_client.reboot_instances(instance_ids: [instance_id])
      puts 'Reboot request sent.'
    end
  end
rescue StandardError => e
  puts "Error requesting reboot: #{e.message}"
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-reboot-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION'
    puts 'Example: ruby ec2-ruby-example-reboot-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)
  request_instance_reboot(ec2_client, instance_id)
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

管理 Amazon EC2 实例

以下代码示例：

1. 停用一个 Amazon EC2 实例。
2. 重新启动实例。
3. 重新启动实例。
4. 启用实例的详细监控。
5. 显示有关可用实例的信息。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example does the following:
# 1. Stops an Amazon Elastic Compute Cloud (Amazon EC2) instance.
# 2. Restarts the instance.
# 3. Reboots the instance.
# 4. Enables detailed monitoring for the instance.
# 5. Displays information about available instances.

require 'aws-sdk-ec2'

# Waits for an Amazon Elastic Compute Cloud (Amazon EC2) instance
# to reach the specified state.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_state [Symbol] The desired instance state.
# @param instance_id [String] The ID of the instance.
# @example
#   wait_for_instance(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     :instance_stopped,
#     'i-033c48ef067af3dEX'
#   )
def wait_for_instance(ec2_client, instance_state, instance_id)
  ec2_client.wait_until(instance_state, instance_ids: [instance_id])
  puts "Success: #{instance_state}."
rescue Aws::Waiters::Errors::WaiterFailed => e
  puts "Failed: #{e.message}"
end

# Attempts to stop an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was stopped; otherwise, false.
# @example
#   exit 1 unless instance_stopped?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
```

```
#   )
def instance_stopped?(ec2_client, instance_id)
  ec2_client.stop_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_stopped, instance_id)
  return true
rescue StandardError => e
  puts "Error stopping instance: #{e.message}"
  return false
end

# Attempts to restart an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was restarted; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_restarted?(ec2_client, instance_id)
  ec2_client.start_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_running, instance_id)
  return true
rescue StandardError => e
  puts "Error restarting instance: #{e.message}"
  return false
end

# Attempts to reboot an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was rebooted; otherwise, false.
# @example
#   exit 1 unless instance_rebooted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_rebooted?(ec2_client, instance_id)
  ec2_client.reboot_instances(instance_ids: [instance_id])
  wait_for_instance(ec2_client, :instance_status_ok, instance_id)
  return true
rescue StandardError => e
  puts "Error rebooting instance: #{e.message}"
  return false
end

# Attempts to enabled detailed monitoring for an
# Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if detailed monitoring was enabled; otherwise, false.
```

```
# @example
#   exit 1 unless instance_detailed_monitoring_enabled?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX'
#   )
def instance_detailed_monitoring_enabled?(ec2_client, instance_id)
  result = ec2_client.monitor_instances(instance_ids: [instance_id])
  puts "Detailed monitoring state: #{result.instance_monitorings[0].monitoring.state}"
  return true
rescue Aws::EC2::Errors::InvalidState
  puts "The instance is not in a monitorable state. Skipping this step."
  return false
rescue StandardError => e
  puts "Error enabling detailed monitoring: #{e.message}"
  return false
end

# Displays information about available
# Amazon Elastic Compute Cloud (Amazon EC2) instances.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_instances_information(Aws::EC2::Client.new(region: 'us-east-1'))
def list_instances_information(ec2_client)
  result = ec2_client.describe_instances
  result.reservations.each do |reservation|
    if reservation.instances.count.positive?
      reservation.instances.each do |instance|
        puts '-' * 12
        puts "Instance ID:               #{instance.instance_id}"
        puts "State:                     #{instance.state.name}"
        puts "Image ID:                  #{instance.image_id}"
        puts "Instance type:             #{instance.instance_type}"
        puts "Architecture:              #{instance.architecture}"
        puts "IAM instance profile ARN:  #{instance.iam_instance_profile.arn}"
        puts "Key name:                  #{instance.key_name}"
        puts "Launch time:               #{instance.launch_time}"
        puts "Detailed monitoring state: #{instance.monitoring.state}"
        puts "Public IP address:         #{instance.public_ip_address}"
        puts "Public DNS name:           #{instance.public_dns_name}"
        puts "VPC ID:                    #{instance.vpc_id}"
        puts "Subnet ID:                 #{instance.subnet_id}"
        if instance.tags.count.positive?
          puts 'Tags:'
          instance.tags.each do |tag|
            puts "                           #{tag.key}/#{tag.value}"
          end
        end
      end
    end
  end
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-manage-instances.rb ' \
      'INSTANCE_ID REGION'
    puts 'Example: ruby ec2-ruby-example-manage-instances.rb ' \
      'i-033c48ef067af3dEX us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
```

```
    instance_id = 'i-033c48ef067af3dEX'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts 'Attempting to stop the instance. ' \
    'This might take a few minutes...'
  unless instance_stopped?(ec2_client, instance_id)
    puts 'Cannot stop the instance. Skipping this step.'
  end

  puts "\nAttempting to restart the instance. " \
    'This might take a few minutes...'
  unless instance_restarted?(ec2_client, instance_id)
    puts 'Cannot restart the instance. Skipping this step.'
  end

  puts "\nAttempting to reboot the instance. " \
    'This might take a few minutes...'
  unless instance_rebooted?(ec2_client, instance_id)
    puts 'Cannot reboot the instance. Skipping this step.'
  end

  puts "\nAttempting to enable detailed monitoring for the instance..."
  unless instance_detailed_monitoring_enabled?(ec2_client, instance_id)
    puts 'Cannot enable detailed monitoring for the instance. ' \
      'Skipping this step.'
  end

  puts "\nInformation about available instances:"
  list_instances_information(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

终止 Amazon EC2 实例

以下示例尝试终止指定的 Amazon EC2 实例。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'

# Attempts to terminate an Amazon Elastic Compute Cloud (Amazon EC2) instance.
#
# Prerequisites:
#
# - The Amazon EC2 instance.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @param instance_id [String] The ID of the instance.
# @return [Boolean] true if the instance was terminated; otherwise, false.
# @example
#   exit 1 unless instance_terminated?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'i-123abc'
#   )
def instance_terminated?(ec2_client, instance_id)
```

```
  response = ec2_client.describe_instance_status(instance_ids: [instance_id])

  if response.instance_statuses.count.positive? &&
    response.instance_statuses[0].instance_state.name == 'terminated'

    puts 'The instance is already terminated.'
    return true
  end

  ec2_client.terminate_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_terminated, instance_ids: [instance_id])
  puts 'Instance terminated.'
  return true
rescue StandardError => e
  puts "Error terminating instance: #{e.message}"
  return false
end

# Full example call:
def run_me
  instance_id = ''
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-terminate-instance-i-123abc.rb ' \
      'INSTANCE_ID REGION '
    puts 'Example: ruby ec2-ruby-example-terminate-instance-i-123abc.rb ' \
      'i-123abc us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    instance_id = 'i-123abc'
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    instance_id = ARGV[0]
    region = ARGV[1]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts "Attempting to terminate instance '#{instance_id}' " \
    '(this might take a few minutes)...'
  unless instance_terminated?(ec2_client, instance_id)
    puts 'Could not terminate instance.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

获取有关 Amazon EC2 区域和可用区的信息

以下示例：

1. 显示可供您AWS 区域使用的 Amazon EC2 列表。
2. 根据Amazon EC2 客户端的不同，显示可供您使用的 Amazon EC2 可用区域列表。AWS 区域

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-ec2'
```

```
# Displays a list of AWS Regions for Amazon Elastic Compute Cloud (Amazon EC2)
# that are available to you.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_regions_endpoints(Aws::EC2::Client.new(region: 'us-east-1'))
def list_regions_endpoints(ec2_client)
  result = ec2_client.describe_regions
  # Enable pretty printing.
  max_region_string_length = 16
  max_endpoint_string_length = 33
  # Print header.
  print 'Region'
  print ' ' * (max_region_string_length - 'Region'.length)
  print "  Endpoint\n"
  print '-' * max_region_string_length
  print '  '
  print '-' * max_endpoint_string_length
  print "\n"
  # Print Regions and their endpoints.
  result.regions.each do |region|
    print region.region_name.to_s
    print ' ' * (max_region_string_length - region.region_name.length)
    print '  '
    print region.endpoint.to_s
    print "\n"
  end
end

# Displays a list of Amazon Elastic Compute Cloud (Amazon EC2)
# Availability Zones available to you depending on the AWS Region
# of the Amazon EC2 client.
#
# @param ec2_client [Aws::EC2::Client] An initialized EC2 client.
# @example
#   list_availability_zones(Aws::EC2::Client.new(region: 'us-east-1'))
def list_availability_zones(ec2_client)
  result = ec2_client.describe_availability_zones
  # Enable pretty printing.
  max_region_string_length = 16
  max_zone_string_length = 18
  max_state_string_length = 9
  # Print header.
  print 'Region'
  print ' ' * (max_region_string_length - 'Region'.length)
  print '  Zone'
  print ' ' * (max_zone_string_length - 'Zone'.length)
  print "  State\n"
  print '-' * max_region_string_length
  print '  '
  print '-' * max_zone_string_length
  print '  '
  print '-' * max_state_string_length
  print "\n"
  # Print Regions, Availability Zones, and their states.
  result.availability_zones.each do |zone|
    print zone.region_name
    print ' ' * (max_region_string_length - zone.region_name.length)
    print '  '
    print zone.zone_name
    print ' ' * (max_zone_string_length - zone.zone_name.length)
    print '  '
    print zone.state
    # Print any messages for this Availability Zone.
    if zone.messages.count.positive?
      print "\n"
```

```
      puts '  Messages for this zone:'
      zone.messages.each do |message|
        print "    #{message.message}\n"
      end
    end
    print "\n"
  end
end

# Full example call:
def run_me
  region = ''
  # Print usage information and then stop.
  if ARGV[0] == '--help' || ARGV[0] == '-h'
    puts 'Usage:   ruby ec2-ruby-example-regions-availability-zones.rb REGION'
    puts 'Example: ruby ec2-ruby-example-regions-availability-zones.rb us-east-1'
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = 'us-east-1'
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  ec2_client = Aws::EC2::Client.new(region: region)

  puts 'AWS Regions for Amazon EC2 that are available to you:'
  list_regions_endpoints(ec2_client)
  puts "\n\nAmazon EC2 Availability Zones that are available to you for AWS Region
 '#{region}':"
  list_availability_zones(ec2_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

AWS Elastic Beanstalk示例：使用AWS适用于 SDK for Ruby

AWS Elastic Beanstalk使您能够迅速在AWS云中部署和管理应用程序，无需为运行这些应用程序的基础设施操心。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 Elastic Beanstalk。有关 Elastic Beanstalk 的更多信息，请参阅文档。AWS Elastic Beanstalk

主题

- 获取有关 AWS Elastic Beanstalk 中的所有应用程序的信息 (p. 85)
- 获取有关 AWS Elastic Beanstalk 中的特定应用程序的信息 (p. 86)
- 更新 AWS Elastic Beanstalk 的 Ruby on Rails 应用程序 (p. 86)

获取有关 AWS Elastic Beanstalk 中的所有应用程序的信息

以下示例列出了您在该区域的所有 Elastic Beanstalk 应用程序的名称、描述和 URL。us-west-2

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk'  # v2: require 'aws-sdk'

eb = Aws::ElasticBeanstalk::Client.new(region: 'us-west-2')

eb.describe_applications.applications.each do |a|
  puts "Name:         #{a.application_name}"
  puts "Description:  #{a.description}"

  eb.describe_environments({application_name: a.application_name}).environments.each do |
env|
    puts "  Environment:  #{env.environment_name}"
    puts "    URL:        #{env.cname}"
    puts "    Health:     #{env.health}"
  end
end
```

获取有关 AWS Elastic Beanstalk 中的特定应用程序的信息

以下示例列出了 MyRailsApp 区域中的 us-west-2 应用程序的名称、描述和 URL。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk'  # v2: require 'aws-sdk'

eb = Aws::ElasticBeanstalk::Client.new(region: 'us-west-2')

app = eb.describe_applications({application_names: [args[0]]})

if app.exists?
  puts "Name:         #{app.application_name}"
  puts "Description:  #{app.description}"

  envs = eb.describe_environments({application_name: app.application_name})
  puts "URL:          #{envs.environments[0].cname}"
end
```

更新 AWS Elastic Beanstalk 的 Ruby on Rails 应用程序

以下示例更新 us-west-2 区域中的 Ruby on Rails 应用程序 MyRailsApp。

> **Note**
>
> 您必须位于 Rails 应用程序的根目录中才能成功运行脚本。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-elasticbeanstalk'  # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

eb = Aws::ElasticBeanstalk::Client.new
s3 = Aws::S3::Client.new

app_name = 'MyRailsApp'

# Get S3 bucket containing app
app_versions = eb.describe_application_versions({ application_name: app_name })
av = app_versions.application_versions[0]
bucket = av.source_bundle.s3_bucket
s3_key = av.source_bundle.s3_key

# Get info on environment
envs = eb.describe_environments({ application_name: app_name })
env = envs.environments[0]
env_name = env.environment_name

# Create new storage location
resp = eb.create_storage_location()

puts "Created storage location in bucket #{resp.s3_bucket}"

s3.list_objects({
  prefix: s3_key,
  bucket: bucket
})

# Create ZIP file
zip_file_basename = SecureRandom.urlsafe_base64.to_s
zip_file_name = zip_file_basename + '.zip'

# Call out to OS to produce ZIP file
cmd = "git archive --format=zip -o #{zip_file_name} HEAD"
%x[ #{cmd} ]

# Get ZIP file contents
zip_contents = File.read(zip_file_name)

key = app_name + "\\" + zip_file_name

s3.put_object({
  body: zip_contents,
  bucket: bucket,
  key: key
})

date = Time.new
today = date.day.to_s + "/" + date.month.to_s + "/" + date.year.to_s

eb.create_application_version({
  process: false,
  application_name: app_name,
```

```
  version_label: zip_file_basename,
  source_bundle: {
    s3_bucket: bucket,
    s3_key: key
  },
  description: "Updated #{today}"
})

eb.update_environment({
  environment_name: env_name,
  version_label: zip_file_basename
})
```

AWS Identity and Access Management(IAM) 使用适用于 Ruby 的AWS开发工具包的示例

AWS Identity and Access Management(IAAAAAAAM) 是一种 Web 服务，用于安全地控制对的访问AWS 服务。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 IAM。有关 IAAAAAAAM 的更多信息，请参阅 IAAAAA AA

主题

- 获取有关 IAM 用户的信息 (p. 88)
- 列出身为管理员的 IAM 用户 (p. 90)
- 添加新的 IAM 用户 (p. 93)
- 为 IAM 用户创建用户访问密钥 (p. 93)
- 向 IAM 用户添加托管策略 (p. 94)
- 创建 IAM 角色 (p. 95)
- 管理 IAM 用户 (p. 97)
- 使用 IAM 策略 (p. 99)
- 管理 IAM 访问密钥 (p. 103)
- 使用 IAM 服务器证书 (p. 106)
- 管理 IAM 账户别名 (p. 109)

获取有关 IAM 用户的信息

以下示例列出了该us-west-2区域中 IAM 用户的群组、策略和访问密钥 ID。如果有 100 个以上的用户，则 iam.list_users.IsTruncated 为 true，并且 iam.list_users.Marker 包含可用于获取有关其他用户的信息的值。请参阅 Aws::IAM::Client.list_users 主题以获取更多信息。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Displays information about available users in
# AWS Identity and Access Management (IAM) including users'
# names, associated group names, inline embedded user policy names,
# and access key IDs.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @example
#   get_user_details(Aws::IAM::Client.new)
def get_user_details(iam_client)
```

```
  users_response = iam_client.list_users

  if users_response.key?('users') && users_response.users.count.positive?

    # Are there more users available than can be displayed?
    if users_response.key?('is_truncated') && users_response.is_truncated
      puts '(Note: not all users are displayed here, ' \
        "only the first #{users_response.users.count}.)"
    else
      puts "Found #{users_response.users.count} user(s):"
    end

    users_response.users.each do |user|
      name = user.user_name
      puts '-' * 30
      puts "User name: #{name}"

      puts "Groups:"
      groups_response = iam_client.list_groups_for_user(user_name: name)
      if groups_response.key?('groups') &&
        groups_response.groups.count.positive?

        groups_response.groups.each do |group|
          puts "  #{group.group_name}"
        end
      else
        puts '  None'
      end

      puts 'Inline embedded user policies:'
      policies_response = iam_client.list_user_policies(user_name: name)
      if policies_response.key?('policy_names') &&
        policies_response.policy_names.count.positive?

        policies_response.policy_names.each do |policy_name|
          puts "  #{policy_name}"
        end
      else
        puts '  None'
      end

      puts 'Access keys:'
      access_keys_response = iam_client.list_access_keys(user_name: name)

      if access_keys_response.key?('access_key_metadata') &&
        access_keys_response.access_key_metadata.count.positive?

        access_keys_response.access_key_metadata.each do |access_key|
          puts "  #{access_key.access_key_id}"
        end
      else
        puts '  None'
      end
    end
  else
    puts 'No users found.'
  end
rescue StandardError => e
  puts "Error getting user details: #{e.message}"
end

# Full example call:
def run_me
  iam_client = Aws::IAM::Client.new
  puts 'Attempting to get details for available users...'
  get_user_details(iam_client)
```

```
end

run_me if $PROGRAM_NAME == __FILE__
```

列出身为管理员的 IAM 用户

以下示例使用 get_account_authorization_details 方法来获取当前账户的用户列表。

选择 Copy 将代码保存在本地。

创建文件 get_admins.rb。

添加所需的 IAM gem 和 os gem，如果你在 Microsoft Windows 上运行，则使用后者使用捆绑的证书。

> **Note**
>
> 适用于 Ruby 的 AWS SDK 的第 2 版没有特定服务的 gem。

```
require 'aws-sdk-iam'  # v2: require 'aws-sdk'
require 'os'

if OS.windows?
 Aws.use_bundled_cert!
end
```

创建一种方法来确定用户是否具有包含管理员权限的策略。

```
def user_has_admin_policy(user, admin_access)
  policies = user.user_policy_list

  policies.each do |p|
    if p.policy_name == admin_access
      return true
    end
  end

  false
end
```

创建一种方法来确定用户是否具有包含管理员权限的附加策略。

```
def user_has_attached_policy(user, admin_access)
  attached_policies = user.attached_managed_policies

  attached_policies.each do |p|
    if p.policy_name == admin_access
      return true
    end
  end

  false
end
```

创建一种方法来确定用户属于的组是否具有包含管理员权限的策略。

```

```

创建一种方法来确定用户属于的组是否具有包含管理员权限的附加策略。

```
def group_has_admin_policy(client, group, admin_access)
 resp = client.list_group_policies(
   group_name: group.group_name
 )

 resp.policy_names.each do |name|
   if name == admin_access
     return true
   end
 end

 false
end
```

创建一种方法来确定用户属于的组是否具有管理员权限。

```
def user_has_admin_from_group(client, user, admin_access)
 resp = client.list_groups_for_user(
   user_name: user.user_name
 )

 resp.groups.each do |group|
   has_admin_policy = group_has_admin_policy(client, group, admin_access)
   if has_admin_policy
     return true
   end

   has_attached_policy = group_has_attached_policy(client, group, admin_access)
   if has_attached_policy
     return true
   end
 end

 false
end
```

创建一种方法来确定用户是否具有管理员权限。

```
def is_user_admin(client, user, admin_access)
 has_admin_policy = user_has_admin_policy(user, admin_access)
 if has_admin_policy
   return true
 end

 has_attached_admin_policy = user_has_attached_policy(user, admin_access)
 if has_attached_admin_policy
   return true
 end

 has_admin_from_group = user_has_admin_from_group(client, user, admin_access)
 if has_admin_from_group
   return true
 end

 false
end
```

创建一种方法来循环访问用户列表并返回其中有多少个用户具有管理员权限。

```
<code>
```

主例程从此处开始。创建 IAM 客户端和变量来存储用户数、具有管理员权限的用户数以及标识提供管理员权限的策略的字符串。

```
def get_admin_count(client, users, admin_access)
 num_admins = 0

 users.each do |user|
   is_admin = is_user_admin(client, user, admin_access)
   if is_admin
     puts user.user_name
     num_admins += 1
   end
 end

 num_admins
end
```

调用 `get_account_authorization_details` 来获取账户的详细信息并从 `user_detail_list` 获取账户的用户。跟踪我们获取了多少用户，调用 `get_admin_count` 来获取具有管理员权限的用户数，并跟踪这些用户的数量。

```
details = client.get_account_authorization_details(
 filter: ['User']
)

users = details.user_detail_list
num_users += users.count
more_admins = get_admin_count(client, users, access_admin)
num_admins += more_admins
```

如果第一个对 `get_account_authorization_details` 的调用没有获取所有详细信息，则再次调用它并重复确定多少用户具有管理员权限的过程。

```
<code>
```

最后，显示多少用户具有管理员权限。

```

more_users = details.is_truncat

而更多_用户
    详情 = client.get_account_authoriz_detail

        过滤器：['用户']，标记：details.marker

    )

    用户 = details.user_detail_list

    num_users += users.count more_admins = get_admin_count（客户端、用户、access_admin）num_admins += more_admins

    more_users = details.is_truncat

结束
```

请参阅上的完整示例GitHub。

添加新的 IAM 用户

以下示例使用密码us-west-2在该区域创建 IAM 用户 my_groovy_userREPLACE_ME，并显示该用户的账户 ID。如果该名称的用户已存在，则会显示一条消息，并且不会创建新用户。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Creates a user in AWS Identity and Access Management (IAM).
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param initial_password [String] The initial password for the user.
# @return [String] The ID of the user if the user was created, otherwise;
#   the string 'Error'.
# @example
#   puts create_user(Aws::IAM::Client.new, 'my-user', 'my-!p@55w0rd!')
def create_user(iam_client, user_name, initial_password)
  response = iam_client.create_user(user_name: user_name)
  iam_client.wait_until(:user_exists, user_name: user_name)
  iam_client.create_login_profile(
    password: initial_password,
    password_reset_required: true,
    user_name: user_name
  )
  return response.user.user_id
rescue Aws::IAM::Errors::EntityAlreadyExists
  puts "Error creating user '#{user_name}': user already exists."
  return 'Error'
rescue StandardError => e
  puts "Error creating user '#{user_name}': #{e.message}"
  return 'Error'
end

# Full example call:
def run_me
  user_name = 'my-user'
  initial_password = 'my-!p@55w0rd!'
  iam_client = Aws::IAM::Client.new

  puts "Attempting to create user '#{user_name}'..."
  user_id = create_user(iam_client, user_name, initial_password)

  if user_id == 'Error'
    puts 'User not created.'
  else
    puts "User '#{user_name}' created with ID '#{user_id}' and initial " \
      "sign-in password '#{initial_password}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

为 IAM 用户创建用户访问密钥

以下示例为该us-west-2地区的 IAM 用户创建访问密钥和私my_groovy_user有密钥。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0
```

```
require 'aws-sdk-iam'

# Creates an access key for a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
#   create_access_key(Aws::IAM::Client.new, 'my-user')
def create_access_key(iam, user_name)
  response = iam.create_access_key(user_name: user_name)
  access_key = response.access_key
  puts 'Access key created:'
  puts "  Access key ID: #{access_key.access_key_id}"
  puts "  Secret access key: #{access_key.secret_access_key}"
  puts 'Keep a record of this information in a secure location. ' \
    'This will be the only time you will be able to view the ' \
    'secret access key.'
rescue Aws::IAM::Errors::LimitExceeded
  puts 'Error creating access key: limit exceeded. Cannot create any more. ' \
    'To create more, delete an existing access key, and then try again.'
rescue StandardError => e
  puts "Error creating access key: #{e.message}"
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  user_name = 'my-user'

  puts 'Attempting to create an access key...'
  create_access_key(iam, user_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

向 IAM 用户添加托管策略

以下示例将托管策略AmazonS3FullAccess添加到该my_groovy_userus-west-2地区的 IAM 用户。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Attaches a policy to a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param policy_arn [String] The Amazon Resource Name (ARN) of the policy.
# @return [Boolean] true if the policy was attached; otherwise, false.
# @example
#   exit 1 unless alias_created?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'arn:aws:iam::aws:policy/AmazonS3FullAccess'
#   )
def policy_attached_to_user?(iam_client, user_name, policy_arn)
  iam_client.attach_user_policy(
```

```
    user_name: user_name,
    policy_arn: policy_arn
  )
  return true
rescue StandardError => e
  puts "Error attaching policy to user: #{e.message}"
  return false
end

# Full example call:
def run_me
  user_name = 'my-user'
  arn_prefix = 'arn:aws:iam::aws:policy/'
  policy_arn = arn_prefix + 'AmazonS3FullAccess'
  iam_client = Aws::IAM::Client.new

  puts "Attempting to attach policy with ARN '#{policy_arn}' to " \
    "user '#{user_name}'..."

  if policy_attached_to_user?(iam_client, user_name, policy_arn)
    puts 'Policy attached.'
  else
    puts 'Policy not attached.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

创建 IAM 角色

以下示例创建了角色，`my_groovy_role`这样 Amazon EC2 就可以访问该地区的 Amazon S3 和 Amazon DynamoDB。`us-west-2`

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-iam'

# Creates a role in AWS Access and Identity Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] A name for the role.
# @param assume_role_policy_document [String]
# @param policy_arns [Array] An array of type String representing
#   Amazon Resource Names (ARNs) corresponding to available
#   IAM managed policies.
# @return [String] The ARN of the new role; otherwise, the string 'Error'.
# @example
#   puts create_role(
#     Aws::IAM::Client.new,
#     'my-ec2-s3-dynamodb-full-access-role',
#     {
#       Version: '2012-10-17',
#       Statement: [
#         {
#           Effect: 'Allow',
#           Principal: {
#             Service: 'ec2.amazonaws.com'
#           },
#           Action: 'sts:AssumeRole'
#         }
#       ]
#     },
#     [
```

```
#       'arn:aws:iam::aws:policy/AmazonS3FullAccess',
#       'arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess'
#     ]
#   )
def create_role(
  iam_client,
  role_name,
  assume_role_policy_document,
  policy_arns
)
  iam_client.create_role(
    role_name: role_name,
    assume_role_policy_document: assume_role_policy_document.to_json
  )
  policy_arns.each do |policy_arn|
    iam_client.attach_role_policy(
      policy_arn: policy_arn,
      role_name: role_name,
    )
  end
  return iam_client.get_role(role_name: role_name).role.arn
rescue StandardError => e
  puts "Error creating role: #{e.message}"
  return 'Error'
end

# Full example call:
def run_me
  role_name = 'my-ec2-s3-dynamodb-full-access-role'

  # Allow the role to trust Amazon Elastic Compute Cloud (Amazon EC2)
  # within the AWS account.
  assume_role_policy_document = {
    Version: '2012-10-17',
    Statement: [
      {
        Effect: 'Allow',
        Principal: {
          Service: 'ec2.amazonaws.com'
        },
        Action: 'sts:AssumeRole'
      }
    ]
  }

  # Allow the role to take all actions within
  # Amazon Simple Storage Service (Amazon S3)
  # and Amazon DynamoDB across the AWS account.
  policy_arns = [
    'arn:aws:iam::aws:policy/AmazonS3FullAccess',
    'arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess'
  ]

  iam_client = Aws::IAM::Client.new

  puts "Attempting to create the role named '#{role_name}'..."

  role_arn = create_role(
    iam_client,
    role_name,
    assume_role_policy_document,
    policy_arns
  )

  if role_arn == 'Error'
    puts 'Could not create role.'
```

```
  else
    puts "Role created with ARN '#{role_arn}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 用户

IAM 用户代表与AWS之交互的个人或服务。有关 IAM 用户的更多信息，请参阅 IAM 用户。

在此示例中，您使用带有 IAM AWS SDK for Ruby 来：

1. 使用 Aws:: AWS IAM:: Client #list_users 获取有关可用 IAM 用户的信息。
2. 通过使用 Aws::IAM::Client#create_user 来创建用户。
3. 通过使用 Aws::IAM::Client#update_user 来更新用户的名称。
4. 通过使用 Aws::IAM::Client#delete_user 来删除用户。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

示例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to to:
# 1. Get a list of user names in AWS Identity and Access Management (IAM).
# 2. Create a user.
# 3. Update the user's name.
# 4. Delete the user.

require 'aws-sdk-iam'

# Gets a list of available user names in
# AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @example
#   list_user_names(Aws::IAM::Client.new)
def list_user_names(iam_client)
  response = iam_client.list_users
  if response.key?('users') && response.users.count.positive?
    response.users.each do |user|
      puts user.user_name
    end
  else
    puts 'No users found.'
  end
rescue StandardError => e
  puts "Error listing user names: #{e.message}"
end

# Creates a user in AWS Identity and Access Management (IAM).
#
```

```
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the new user.
# @return [Boolean] true if the user was created; otherwise, false.
# @example
#   exit 1 unless user_created?(Aws::IAM::Client.new, 'my-user')
def user_created?(iam_client, user_name)
  iam_client.create_user(user_name: user_name)
  return true
rescue Aws::IAM::Errors::EntityAlreadyExists
  puts "Error creating user: user '#{user_name}' already exists."
  return false
rescue StandardError => e
  puts "Error creating user: #{e.message}"
  return false
end

# Changes the name of a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_current_name [String] The current name of the user.
# @param user_new_name [String] The new name for the user.
# @return [Boolean] true if the name of the user was changed;
#   otherwise, false.
# @example
#   exit 1 unless user_name_changed?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'my-changed-user'
#   )
def user_name_changed?(iam_client, user_current_name, user_new_name)
  iam_client.update_user(
    user_name: user_current_name,
    new_user_name: user_new_name
  )
  return true
rescue StandardError => e
  puts "Error updating user name: #{e.message}"
  return false
end

# Deletes a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @return [Boolean] true if the user was deleted; otherwise, false.
# @example
#   exit 1 unless user_deleted?(Aws::IAM::Client.new, 'my-user')
def user_deleted?(iam_client, user_name)
  iam_client.delete_user(user_name: user_name)
  return true
rescue StandardError => e
  puts "Error deleting user: #{e.message}"
  return false
end

# Full example call:
def run_me
  user_name = 'my-user'
  user_changed_name = 'my-changed-user'
  delete_user = true
```

```
  iam_client = Aws::IAM::Client.new

  puts "Initial user names are:\n\n"
  list_user_names(iam_client)

  puts "\nAttempting to create user '#{user_name}'..."

  if user_created?(iam_client, user_name)
    puts 'User created.'
  else
    puts 'Could not create user. Stopping program.'
    exit 1
  end

  puts "User names now are:\n\n"
  list_user_names(iam_client)

  puts "\nAttempting to change the name of the user '#{user_name}' " \
    "to '#{user_changed_name}'..."

  if user_name_changed?(iam_client, user_name, user_changed_name)
    puts 'User name changed.'
    puts "User names now are:\n\n"
    list_user_names(iam_client)

    if delete_user
      # Delete user with changed name.
      puts "\nAttempting to delete user '#{user_changed_name}'..."

      if user_deleted?(iam_client, user_changed_name)
        puts 'User deleted.'
      else
        puts 'Could not delete user. You must delete the user yourself.'
      end

      puts "User names now are:\n\n"
      list_user_names(iam_client)
    end
  else
    puts 'Could not change user name.'
    puts "User names now are:\n\n"
    list_user_names(iam_client)

    if delete_user
      # Delete user with initial name.
      puts "\nAttempting to delete user '#{user_name}'..."

      if user_deleted?(iam_client, user_name)
        puts 'User deleted.'
      else
        puts 'Could not delete user. You must delete the user yourself.'
      end

      puts "User names now are:\n\n"
      list_user_names(iam_client)
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 IAM 策略

IAM 策略是指定一项或多项权限的文档。有关 IAM 策略的更多信息，请参阅 IAM 策略概述。

在此示例中，您使用带有 IAM AWS SDK for Ruby 来：

1. 使用 Aws::IAM::Client#create_policy 创建策略。
2. 使用 Aws::IAM::Client#get_policy 获取有关策略的信息。
3. 使用 Aws::IAM::Client#attach_role_policy 将策略附加到角色。
4. 使用 Aws::IAM::Client#list_attached_role_policies 列出附加到角色的策略。
5. 使用 Aws::IAM::Client#detach_role_policy 从角色分离策略。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要创建脚本中指定的角色 (my-role)。您可以在 IAM 控制台中执行此操作。

示例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to:
# 1. Create a policy in AWS Identity and Access Management (IAM).
# 2. Attach the policy to a role.
# 3. List the policies that are attached to the role.
# 4. Detach the policy from the role.

require 'aws-sdk-iam'

# Creates a policy in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param policy_name [String] A name for the policy.
# @param policy_document [Hash] The policy definition.
# @return [String] The new policy's Amazon Resource Name (ARN);
#   otherwise, the string 'Error'.
# @example
#   puts create_policy(
#     Aws::IAM::Client.new,
#     'my-policy',
#     {
#       'Version': '2012-10-17',
#       'Statement': [
#         {
#           'Effect': 'Allow',
#           'Action': 's3:ListAllMyBuckets',
#           'Resource': 'arn:aws:s3:::*'
#         }
#       ]
#     }
#   )
def create_policy(iam_client, policy_name, policy_document)
  response = iam_client.create_policy(
    policy_name: policy_name,
    policy_document: policy_document.to_json
  )
  return response.policy.arn
rescue StandardError => e
```

```
  puts "Error creating policy: #{e.message}"
  return 'Error'
end

# Attaches a policy to a role in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - An existing role.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to attach the policy to.
# @param policy_arn [String] The policy's Amazon Resource Name (ARN).
# @return [Boolean] True if the policy was attached to the role;
#   otherwise, false.
# @example
#   exit 1 unless policy_attached_to_role?(
#     Aws::IAM::Client.new,
#     'my-role',
#     'arn:aws:iam::111111111111:policy/my-policy'
#   )
def policy_attached_to_role?(iam_client, role_name, policy_arn)
  iam_client.attach_role_policy(role_name: role_name, policy_arn: policy_arn)
  return true
rescue StandardError => e
  puts "Error attaching policy to role: #{e.message}"
  return false
end

# Displays a list of policy Amazon Resource Names (ARNs) that are attached to a
# role in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - An existing role.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role.
# @example
#   list_policy_arns_attached_to_role(Aws::IAM::Client.new, 'my-role')
def list_policy_arns_attached_to_role(iam_client, role_name)
  response = iam_client.list_attached_role_policies(role_name: role_name)
  if response.key?('attached_policies') && response.attached_policies.count.positive?
    response.attached_policies.each do |attached_policy|
      puts "  #{attached_policy.policy_arn}"
    end
  else
    puts 'No policies attached to role.'
  end
rescue StandardError => e
  puts "Error checking for policies attached to role: #{e.message}"
end

# Detaches a policy from a role in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - An existing role with an attached policy.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to detach the policy from.
# @param policy_arn [String] The policy's Amazon Resource Name (ARN).
# @return [Boolean] True if the policy was detached from the role;
#   otherwise, false.
# @example
#   exit 1 unless policy_detached_from_role?(
#     Aws::IAM::Client.new,
#     'my-role',
#     'arn:aws:iam::111111111111:policy/my-policy'
```

```
#   )
def policy_detached_from_role?(iam_client, role_name, policy_arn)
  iam_client.detach_role_policy(role_name: role_name, policy_arn: policy_arn)
  return true
rescue StandardError => e
  puts "Error detaching policy from role: #{e.message}"
  return false
end

# Full example call:
def run_me
  role_name = 'my-role'
  policy_name = 'my-policy'

  # Allows the caller to get a list of all buckets in
  # Amazon Simple Storage Service (Amazon S3) that are owned by the caller.
  policy_document = {
    'Version': '2012-10-17',
    'Statement': [
      {
        'Effect': 'Allow',
        'Action': 's3:ListAllMyBuckets',
        'Resource': 'arn:aws:s3:::*'
      }
    ]
  }

  detach_policy_from_role = true
  iam_client = Aws::IAM::Client.new

  puts "Attempting to create policy '#{policy_name}'..."
  policy_arn = create_policy(iam_client, policy_name, policy_document)

  if policy_arn == 'Error'
    puts 'Could not create policy. Stopping program.'
    exit 1
  else
    puts 'Policy created.'
  end

  puts "Attempting to attach policy '#{policy_name}' " \
    "to role '#{role_name}'..."

  if policy_attached_to_role?(iam_client, role_name, policy_arn)
    puts 'Policy attached.'
  else
    puts 'Could not attach policy to role.'
    detach_policy_from_role = false
  end

  puts "Policy ARNs attached to role '#{role_name}':"
  list_policy_arns_attached_to_role(iam_client, role_name)

  if detach_policy_from_role
    puts "Attempting to detach policy '#{policy_name}' " \
      "from role '#{role_name}'..."

    if policy_detached_from_role?(iam_client, role_name, policy_arn)
      puts 'Policy detached.'
    else
      puts 'Could not detach policy from role. You must detach it yourself.'
    end

  end
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 访问密钥

用户需要自己的访问密钥才能通过 AWS SDK for Ruby 进行编程调用。AWS要满足这一需要，您可以创建、修改、查看或轮换 IAM 用户的访问密钥 (访问密钥 ID 和秘密访问密钥)。默认情况下，当您创建访问密钥时，其状态为“Active”(活动)。这表示用户可以将访问密钥用于 API 调用。有关 访问密钥的更多信息，请参阅管理 IAM 用户的访问密钥。

在此示例中，您使用带有 IAM AWS SDK for Ruby 来：

1. 使用 Aws:: AWS IAM:: Client #list_access_keys 列出 IAM 用户访问密钥。
2. 使用 Aws::IAM::Client#create_access_key 创建访问密钥。
3. 使用 Aws::IAM::Client#get_access_key_last_used 确定访问密钥的上次使用时间。
4. 使用 Aws::IAM::Client#update_access_key 停用访问密钥。
5. 使用 Aws::IAM::Client#delete_access_key 删除访问密钥。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要创建脚本中指定的用户 (my-user)。您可以在 IAM 控制台中或以编程方式创建新 IAM 用户，如添加新 IAM 用户 (p. 93)中所示。

示例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# This code example demonstrates how to:
# 1. List access keys for a user in AWS Identity and Access Management (IAM).
# 2. Create an access key for a user.
# 3. Determine when a user's access keys were last used.
# 4. Deactivate an access key for a user.
# 5. Delete an access key for a user.

require 'aws-sdk-iam'

# Lists information about access keys for a user in
# AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
#   puts list_access_keys(Aws::IAM::Client.new, 'my-user')
def list_access_keys(iam, user_name)
  response = iam.list_access_keys(user_name: user_name)

  if response.access_key_metadata.count.positive?
    puts 'Access key IDs:'
    response.access_key_metadata.each do |key_metadata|
      puts "  #{key_metadata.access_key_id}"
```

```
    end
  else
    puts "No access keys found for user '#{user_name}'."
  end
rescue Aws::IAM::Errors::NoSuchEntity
  puts "Error listing access keys: cannot find user '#{user_name}'."
  exit 1
rescue StandardError => e
  puts "Error listing access keys: #{e.message}"
end

# Creates an access key for a user in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @return [Aws::IAM::Types::AccessKey] Information about the new access key;
#   otherwise, the string 'Error'.
# @example
#   puts create_access_key(Aws::IAM::Client.new, 'my-user')
def create_access_key(iam, user_name)
  response = iam.create_access_key(user_name: user_name)
  access_key = response.access_key
  puts 'Access key created:'
  puts "  Access key ID: #{access_key.access_key_id}"
  puts "  Secret access key: #{access_key.secret_access_key}"
  puts 'Keep a record of this information in a secure location. ' \
    'This will be the only time you will be able to view the ' \
    'secret access key.'
  return access_key
rescue Aws::IAM::Errors::LimitExceeded
  puts 'Error creating access key: limit exceeded. Cannot create any more. ' \
    'To create more, delete an existing access key, and then try again.'
  return 'Error'
rescue StandardError => e
  puts "Error creating access key: #{e.message}"
  return 'Error'
end

# Lists information about when access keys for a user in
# AWS Identity and Access Management (IAM) were last used.
#
# Prerequisites:
# - The user in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @example
#   puts access_keys_last_used(Aws::IAM::Client.new, 'my-user')
def access_keys_last_used(iam, user_name)
  response = iam.list_access_keys(user_name: user_name)

  response.access_key_metadata.each do |key_metadata|
    last_used = iam.get_access_key_last_used(access_key_id: key_metadata.access_key_id)
    if last_used.access_key_last_used.last_used_date.nil?
      puts "  Key '#{key_metadata.access_key_id}' not used or date undetermined."
    else
      puts "  Key '#{key_metadata.access_key_id}' last used on " \
      "#{last_used.access_key_last_used.last_used_date}"
    end
  end
rescue StandardError => e
  puts "Error determining when access keys were last used: #{e.message}"
end
```

```
# Deactivates an access key in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - A user in IAM.
# - An access key for that user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID of the access key.
# @return [Boolean] true if the access key was deactivated;
#   otherwise, false.
# @example
#   exit 1 unless access_key_deactivated?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'AKIAIOSFODNN7EXAMPLE'
#   )
def access_key_deactivated?(iam, user_name, access_key_id)
  iam.update_access_key(
    user_name: user_name,
    access_key_id: access_key_id,
    status: 'Inactive'
  )
  return true
rescue StandardError => e
  puts "Error deactivating access key: #{e.message}"
  return false
end

# Deletes an access key in AWS Identity and Access Management (IAM).
#
# Prerequisites:
# - A user in IAM.
# - An access key for that user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The name of the user.
# @param access_key_id [String] The ID of the access key.
# @return [Boolean] true if the access key was deleted;
#   otherwise, false.
# @example
#   exit 1 unless access_key_deleted?(
#     Aws::IAM::Client.new,
#     'my-user',
#     'AKIAIOSFODNN7EXAMPLE'
#   )
def access_key_deleted?(iam, user_name, access_key_id)
  iam.delete_access_key(
    user_name: user_name,
    access_key_id: access_key_id
  )
  return true
rescue StandardError => e
  puts "Error deleting access key: #{e.message}"
  return false
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  user_name = 'my-user'
  create_key = true  # Set to false to not create a new access key.
  delete_key = true  # Set to false to not delete any generated access key.

  puts "Access keys for user '#{user_name}' before attempting to create an " \
```

```
    'additional access key for the user:'
  list_access_keys(iam, user_name)

  access_key = ''

  if create_key
    puts 'Attempting to create an additional access key...'
    access_key = create_access_key(iam, user_name)

    if access_key == 'Error'
      puts 'Additional access key not created. Stopping program.'
      exit 1
    end

    puts 'Additional access key created. Access keys for user now are:'
    list_access_keys(iam, user_name)
  end

  puts 'Determining when current access keys were last used...'
  access_keys_last_used(iam, user_name)

  if create_key && delete_key
    puts 'Attempting to deactivate additional access key...'

    if access_key_deactivated?(iam, user_name, access_key.access_key_id)
      puts 'Access key deactivated. Access keys for user now are:'
      list_access_keys(iam, user_name)
    else
      puts 'Access key not deactivated. Stopping program.'
      puts 'You will need to delete the access key yourself.'
    end

    puts 'Attempting to delete additional access key...'

    if access_key_deleted?(iam, user_name, access_key.access_key_id)
      puts 'Access key deleted. Access keys for user now are:'
      list_access_keys(iam, user_name)
    else
      puts 'Access key not deleted. You will need to delete the ' \
        'access key yourself.'
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 IAM 服务器证书

要在上启用与您的网站或应用程序的 HTTPS 连接AWS，您需要 SSL/TLS 服务器证书。要在您的网站或应用程序上AWS使用您从外部提供商那里获得的证书，您必须将证书上传到 IAM 或将其导入到Certificate Manager 中AWS。有关服务器证书的更多信息，请参阅使用服务器证书。

在此示例中，您使用带有 IAM AWS SDK for Ruby 来：

1. 使用 Aws::IAM::Client#update_server_certificate 更新服务器证书。
2. 使用 Aws::IAM::Client#delete_server_certificate 删除服务器证书。
3. 使用 Aws::IAM::Client#list_server_certificates 列出有关任何剩余的服务器证书的信息。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

Note

服务器证书必须已经存在，否则脚本将抛出 Aws:: IAM:: NoSuchEntity 错误。

示例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

# The following code example shows how to:
# 1. Update a server certificate in AWS Identity and Access Management (IAM).
# 2. List the names of available server certificates.
# 3. Delete a server certificate.

require 'aws-sdk-iam'

# Gets a list of available server certificate names in
# AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @example
#   list_server_certificate_names(Aws::IAM::Client.new)
def list_server_certificate_names(iam_client)
  response = iam_client.list_server_certificates

  if response.key?('server_certificate_metadata_list') &&
    response.server_certificate_metadata_list.count.positive?

    response.server_certificate_metadata_list.each do |certificate_metadata|
      puts certificate_metadata.server_certificate_name
    end
  else
    puts 'No server certificates found. Stopping program.'
    exit 1
  end
rescue StandardError => e
  puts "Error getting server certificate names: #{e.message}"
end

# Changes the name of a server certificate in
# AWS Identity and Access Management (IAM).
#
# Prerequisites:
#
# - The server certificate in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param server_certificate_current_name [String] The current name of
#   the server certificate.
# @param server_certificate_new_name [String] The new name for the
#   the server certificate.
# @return [Boolean] true if the name of the server certificate
#   was changed; otherwise, false.
# @example
#   exit 1 unless server_certificate_name_changed?(
#     Aws::IAM::Client.new,
#     'my-server-certificate',
#     'my-changed-server-certificate'
#   )
def server_certificate_name_changed?(
```

```
  iam_client,
  server_certificate_current_name,
  server_certificate_new_name
)
  iam_client.update_server_certificate(
    server_certificate_name: server_certificate_current_name,
    new_server_certificate_name: server_certificate_new_name
  )
  return true
rescue StandardError => e
  puts "Error updating server certificate name: #{e.message}"
  return false
end

# Deletes a server certificate in
# AWS Identity and Access Management (IAM).
#
# Prerequisites:
#
# - The server certificate in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param server_certificate_name [String] The name of the server certificate.
# @return [Boolean] true if the server certificate was deleted;
#   otherwise, false.
# @example
#   exit 1 unless server certificate_deleted?(
#     Aws::IAM::Client.new,
#     'my-server-certificate'
#   )
def server_certificate_deleted?(iam_client, server_certificate_name)
  iam_client.delete_server_certificate(
    server_certificate_name: server_certificate_name
  )
  return true
rescue StandardError => e
  puts "Error deleting server certificate: #{e.message}"
  return false
end

# Full example call:
def run_me
  server_certificate_name = 'my-server-certificate'
  server_certificate_changed_name = 'my-changed-server-certificate'
  delete_server_certificate = true
  iam_client = Aws::IAM::Client.new

  puts "Initial server certificate names are:\n\n"
  list_server_certificate_names(iam_client)

  puts "\nAttempting to change name of server certificate " \
    " '#{server_certificate_name}' " \
    "to '#{server_certificate_changed_name}'..."

  if server_certificate_name_changed?(
    iam_client,
    server_certificate_name,
    server_certificate_changed_name
  )
    puts 'Server certificate name changed.'
    puts "Server certificate names now are:\n\n"
    list_server_certificate_names(iam_client)

    if delete_server_certificate
      # Delete server certificate with changed name.
      puts "\nAttempting to delete server certificate " \
```

```
        "'#{server_certificate_changed_name}'..."

      if server_certificate_deleted?(iam_client, server_certificate_changed_name)
        puts 'Server certificate deleted.'
      else
        puts 'Could not delete server certificate. You must delete it yourself.'
      end

      puts "Server certificate names now are:\n\n"
      list_server_certificate_names(iam_client)
    end
  else
    puts 'Could not change server certificate name.'
    puts "Server certificate names now are:\n\n"
    list_server_certificate_names(iam_client)

    if delete_server_certificate
      # Delete server certificate with initial name.
      puts "\nAttempting to delete server certificate '#{server_certificate_name}'..."

      if server_certificate_deleted?(iam_client, server_certificate_name)
        puts 'Server certificate deleted.'
      else
        puts 'Could not delete server certificate. You must delete it yourself.'
      end

      puts "Server certificate names now are:\n\n"
      list_server_certificate_names(iam_client)
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 IAM 账户别名

如果您希望登录页面的 URL 用贵公司名称或其他友好标识符取代您的AWS账户 ID，可以为您的账户 ID 创建 IAM AWS 账户别名。如果创建 IAAAAAAAAM 账户别名，您的登录页面 URL 地址格式会更改为包含该别名。有关 IAAAAAAAAM 账户别名的更多信息，请参阅AWS账户 ID 及其别名。

在此示例中，您使用带有 IAM AWS SDK for Ruby 来：

1. 使用 Aws:: IAM:: Client #list_account_aliases 列出AWS账户别名。
2. 使用 Aws::IAM::Client#create_account_alias 创建账户别名。
3. 使用 Aws::IAM::Client#delete_account_alias 删除账户别名。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

在示例代码中，将my-account-alias字符串更改为在所有 Amazon Web Services 产品中是唯一的名称。

示例

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0
```

```
# The following code example shows how to:
# 1. List available AWS account aliases.
# 2. Create an account alias.
# 3. Delete an account alias.

require 'aws-sdk-iam'

# Lists available AWS account aliases.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @example
#   puts list_aliases(Aws::IAM::Client.new)
def list_aliases(iam)
  response = iam.list_account_aliases

  if response.account_aliases.count.positive?
    response.account_aliases.each do |account_alias|
      puts "  #{account_alias}"
    end
  else
    puts 'No account aliases found.'
  end
rescue StandardError => e
  puts "Error listing account aliases: #{e.message}"
end

# Creates an AWS account alias.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param account_alias [String] The name of the account alias to create.
# @return [Boolean] true if the account alias was created; otherwise, false.
# @example
#   exit 1 unless alias_created?(Aws::IAM::Client.new, 'my-account-alias')
def alias_created?(iam, account_alias)
  iam.create_account_alias(account_alias: account_alias)
  return true
rescue StandardError => e
  puts "Error creating account alias: #{e.message}"
  return false
end

# Deletes an AWS account alias.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param account_alias [String] The name of the account alias to delete.
# @return [Boolean] true if the account alias was deleted; otherwise, false.
# @example
#   exit 1 unless alias_deleted?(Aws::IAM::Client.new, 'my-account-alias')
def alias_deleted?(iam, account_alias)
  iam.delete_account_alias(account_alias: account_alias)
  return true
rescue StandardError => e
  puts "Error deleting account alias: #{e.message}"
  return false
end

# Full example call:
def run_me
  iam = Aws::IAM::Client.new
  account_alias = 'my-account-alias'
  create_alias = true # Change to false to not generate an account alias.
  delete_alias = true # Change to false to not delete any generated account alias.

  puts 'Account aliases are:'
  list_aliases(iam)
```

```
  if create_alias
    puts 'Attempting to create account alias...'
    if alias_created?(iam, account_alias)
      puts 'Account alias created. Account aliases now are:'
      list_aliases(iam)
    else
      puts 'Account alias not created. Stopping program.'
      exit 1
    end
  end

  if create_alias && delete_alias
    puts 'Attempting to delete account alias...'
    if alias_deleted?(iam, account_alias)
      puts 'Account alias deleted. Account aliases now are:'
      list_aliases(iam)
    else
      puts 'Account alias not deleted. You will need to delete ' \
        'the alias yourself.'
    end
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

AWS Key Management Service示例：使用AWS适用于SDK for Ruby

AWS Key Management Service (AWS KMS) 是可扩展到云的加密和密钥管理服务。您可以使用以下示例使用适用于 Ruby 的 AWS SDK 进行访问AWS KMS。有关 AWS KMS 的详细信息，请参阅 AWS KMS 文档。有关 AWS KMS 客户端的参考信息，请参阅 Aws::KMS::Client。

主题

- 创建 AWS KMS key (p. 111)
- 在 AWS KMS 中对数据进行加密 (p. 112)
- 在 AWS KMS 中对数据 Blob 进行解密 (p. 112)
- 在 AWS KMS 中对数据 Blob 重新加密 (p. 113)

创建 AWS KMS key

以下示例使用 AWS SDK for Ruby create_key 方法，该方法实现了创建的CreateKey操作。AWS KMS keys 由于该示例仅加密少量数据，因此 KMS 密钥可以满足我们的目的。对于更大的数据量，使用 KMS 密钥加密数据加密密钥 (DEK)。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
                           tags: [
```

```
                            {
                              tag_key: "CreatedBy",
                              tag_value: "ExampleUser"
                            }
                          ]
                        })

puts resp.key_metadata.key_id
```

请参阅上的完整示例GitHub。

在 AWS KMS 中对数据进行加密

以下示例使用实现加密操作的 AWS SDK for Ruby 加密方法对字符串 “1234567890” 进行加密。此示例显示了生成的加密 Blob 的可读版本。

```
require "aws-sdk-kms"  # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

keyId = "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"

text = "1234567890"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.encrypt({
  key_id: keyId,
  plaintext: text,
})

# Display a readable version of the resulting encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

请参阅上的完整示例GitHub。

在 AWS KMS 中对数据 Blob 进行解密

以下示例使用 AWS SDK for Ruby 解密方法（该方法实现了解密操作）来解密提供的字符串并发出结果。

```
require "aws-sdk-kms" # v2: require 'aws-sdk'

# Decrypted blob

blob =
 "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596e093d4ffbf2
blob_packed = [blob].pack("H*")

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.decrypt({
                        ciphertext_blob: blob_packed
                      })

puts "Raw text: "
puts resp.plaintext
```

请参阅上的完整示例GitHub。

在 AWS KMS 中对数据 Blob 重新加密

以下示例使用实现ReEncrypt操作的 AWS SDK for Ruby re_enc rypt 方法来解密加密数据，然后立即在新版本下重新加密数据。AWS KMS key这些操作全部都在 AWS KMS 内的服务器端执行，因此它们永远不会将您的明文在 AWS KMS 外公开。此示例显示了生成的重新加密 Blob 的可读版本。

```
require "aws-sdk-kms"  # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
 "01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593596e093d4ffbf2
sourceCiphertextBlob = [blob].pack("H*")

# Replace the fictitious key ARN with a valid key ID

destinationKeyId = "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"

client = Aws::KMS::Client.new(region: "us-west-2")

resp = client.re_encrypt({
  ciphertext_blob: sourceCiphertextBlob,
  destination_key_id: destinationKeyId
})

# Display a readable version of the resulting re-encrypted blob.
puts "Blob:"
puts resp.ciphertext_blob.unpack("H*")
```

请参阅上的完整示例GitHub。

AWS Lambda示例：使用AWS适用于 SDK for Ruby

AWS Lambda(Lambda) 是一个面向后端 Web 开发人员的零管理计算平台，可在AWS云端为您运行代码，并为您提供精细的定价结构。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 Lambda。有关 Lambda 的更多信息，请参阅AWS Lambda文档。

主题

- 显示有关所有 Lambda 函数的信息 (p. 113)
- 创建 Lambda 函数 (p. 114)
- 运行 Lambda 函数 (p. 115)
- 配置 Lambda 函数以接收通知 (p. 116)

显示有关所有 Lambda 函数的信息

以下示例显示了您在该us-west-2区域的所有 Lambda 函数的名称、ARN 和角色。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda'  # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')

client.list_functions.functions.each do |function|
  puts 'Name: ' + function.function_name
  puts 'ARN:  ' + function.function_arn
  puts 'Role: ' + function.role
  puts
end
```

创建 Lambda 函数

以下示例使用这些值创建了在us-west-2区域my-notification-function中命名的 Lambda 函数：

- 角色 ARN: my-resource-arn。在大多数情况下，您只需将 AWSLambdaExecute 托管策略附加到此角色的策略。
- 函数入口点：my-package.my-class
- Runtime: java8
- Zip 文件：my-zip-file.zip
- 存储桶: my-notification-bucket
- 键：my-zip-file

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda'  # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')

args = {}
args[:role] = 'my-resource-arn'
args[:function_name] = 'my-notification-function'
args[:handler] = 'my-package.my-class'

# Also accepts nodejs, nodejs4.3, and python2.7
args[:runtime] = 'java8'

code = {}
code[:zip_file] = 'my-zip-file.zip'
code[:s3_bucket] = 'my-notification-bucket'
code[:s3_key] = 'my-zip-file'

args[:code] = code
```

```
client.create_function(args)
```

运行 Lambda 函数

以下示例运行在us-west-2区域MyGetitemsFunction中命名的 Lambda 函数。此函数从数据库中返回项目的列表。输入 JSON 如下所示。

```
{
   "SortBy": "name|time",
   "SortOrder": "ascending|descending",
   "Number": 50
}
```

其中：

- SortBy 是用于对结果进行排序的标准。我们的示例使用 time，这意味着返回的项目将按照它们添加到数据库中的顺序排序。
- SortOrder 是排序顺序。我们的示例使用 descending，这意味着最新的项目位于列表的最后。
- Number 是要检索的项目的最大数量 (默认值为 50)。我们的示例使用 10，这意味着获取 10 个最新项目。

输出 JSON 如下所示，其中：

- STATUS-CODE 是 HTTP 状态代码，200 意味着调用成功。
- RESULT 是调用的结果，即 success 或 failure。
- 如果 ERROR 是 result，则 failure 是错误消息，否则是空字符串
- 如果 DATA 为 result，则 success 为返回的结果数组，否则为 nil。

```
{
   "statusCode": "STATUS-CODE",
   "body": {
      "result": "RESULT",
      "error": "ERROR",
      "data": "DATA"
   }
}
```

第一步是加载我们使用的模块：

- aws-sdk加载我们用来调用 Lambda 函数的 AWS SDK for Ruby 模块。
- json 加载我们用于封送和解组请求和响应负载的 JSON 模块。
- os 加载我们用于确保我们可以在 Microsoft Windows 上运行 Ruby 应用程序的操作系统模块。如果您使用的是不同的操作系统，可以删除这些行。
- 然后我们创建 Lambda 客户端，用于调用 Lambda 函数。
- 接下来，我们为请求参数创建哈希并调用 MyGetItemsFunction。
- 最后，我们解析响应，如果成功，我们会输出项目。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda'  # v2: require 'aws-sdk'
require 'json'

# To run on Windows:
require 'os'
if OS.windows?
  Aws.use_bundled_cert!
end

client = Aws::Lambda::Client.new(region: 'us-west-2')

# Get the 10 most recent items
req_payload = {:SortBy => 'time', :SortOrder => 'descending', :NumberToGet => 10}
payload = JSON.generate(req_payload)

resp = client.invoke({
                       function_name: 'MyGetItemsFunction',
                       invocation_type: 'RequestResponse',
                       log_type: 'None',
                       payload: payload
                     })

resp_payload = JSON.parse(resp.payload.string) # , symbolize_names: true)

# If the status code is 200, the call succeeded
if resp_payload["statusCode"] == 200
  # If the result is success, we got our items
  if resp_payload["body"]["result"] == "success"
    # Print out items
    resp_payload["body"]["data"].each do |item|
      puts item
    end
  end
end
```

请参阅上的完整示例GitHub。

配置 Lambda 函数以接收通知

以下示例将us-west-2区域my-notification-function中命名的 Lambda 函数配置为接受来自 ARN 的资源的通知。my-resource-arn

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-lambda'  # v2: require 'aws-sdk'

client = Aws::Lambda::Client.new(region: 'us-west-2')
```

```
args = {}
args[:function_name] = 'my-notification-function'
args[:statement_id] = 'lambda_s3_notification'
args[:action] = 'lambda:InvokeFunction'
args[:principal] = 's3.amazonaws.com'
args[:source_arn] = 'my-resource-arn'

client.add_permission(args)
```

使用适用于 Ruby 的 AWS SDK 的Amazon Polly 示例

Amazon Polly 云服务可以将文本转化为逼真的语音。AWSSDK for Ruby 示例可以将Amazon Polly 集成到您的应用程序中。要了解有关 Amazon Polly 的更多信息，请参阅 Amazon Polly 文档。示例假定您已设置并配置开发工具包（即您已导入所有必需的软件包并已设置您的凭证和区域）。有关更多信息，请参阅安装适用于 Ruby 的AWS软件开发工具包和配置适用于 Rub AWS (p. 3) SDK for Ruby (p. 9)。

主题

- 获取语音列表 (p. 117)
- 获取词典列表 (p. 118)
- 合成语音 (p. 119)

获取语音列表

此示例使用 describe_voices 方法获取 us-west-2 区域中的美国英语语音列表。

选择 Copy 将代码保存在本地。

创建文件 polly_describe_voices.rb。

添加必需的 Gem。

> **Note**
>
> 适用于 Ruby 的 AWS SDK 的第 2 版没有特定服务的 gem。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly'  # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  # Get US English voices
  resp = polly.describe_voices(language_code: 'en-US')
```

```
  resp.voices.each do |v|
    puts v.name
    puts '  ' + v.gender
    puts
  end
rescue StandardError => ex
  puts 'Could not get voices'
  puts 'Error message:'
  puts ex.message
end
```

请参阅上的完整示例GitHub。

获取词典列表

此示例使用 list_lexicons 方法获取 us-west-2 区域中的词典列表。

选择 Copy 将代码保存在本地。

创建文件 polly_list_lexicons.rb。

添加必需的 Gem。

> **Note**
>
> 适用于 Ruby 的 AWS SDK 的第 2 版没有特定服务的 gem。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly'  # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  resp = polly.list_lexicons

  resp.lexicons.each do |l|
    puts l.name
    puts '  Alphabet:' + l.attributes.alphabet
    puts '  Language:' + l.attributes.language
    puts
  end
rescue StandardError => ex
  puts 'Could not get lexicons'
  puts 'Error message:'
  puts ex.message
end
```

请参阅上的完整示例GitHub。

合成语音

此示例使用 synthesize_speech 方法从文件中获取文本并生成包含合成的语音的 MP3 文件。

选择 Copy 将代码保存在本地。

创建文件 polly_synthesize_speech.rb。

添加必需的 Gem。

> **Note**
>
> 适用于 Ruby 的 AWS SDK 的第 2 版没有特定服务的 gem。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-polly'  # In v2: require 'aws-sdk'

begin
  # Get the filename from the command line
  if ARGV.empty?()
    puts 'You must supply a filename'
    exit 1
  end

  filename = ARGV[0]

  # Open file and get the contents as a string
  if File.exist?(filename)
    contents = IO.read(filename)
  else
    puts 'No such file: ' + filename
    exit 1
  end

  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/config
  polly = Aws::Polly::Client.new

  resp = polly.synthesize_speech({
    output_format: "mp3",
    text: contents,
    voice_id: "Joanna",
  })

  # Save output
  # Get just the file name
  #  abc/xyz.txt -> xyx.txt
  name = File.basename(filename)

  # Split up name so we get just the xyz part
  parts = name.split('.')
  first_part = parts[0]
```

```
  mp3_file = first_part + '.mp3'

  IO.copy_stream(resp.audio_stream, mp3_file)

  puts 'Wrote MP3 content to: ' + mp3_file
rescue StandardError => ex
  puts 'Got error:'
  puts 'Error message:'
  puts ex.message
end
```

Note

生成的 MP3 文件采用 MPEG-2 格式。

请参阅上的完整示例GitHub。

亚马逊 RDS 示例使用适用于 R AWS SDK for Ruby

Amazon Relational Database Service ase Database Service (Amazon RDS) 是一项 Web 服务，让用户能够在云中更轻松地设置、操作和扩展关系数据库。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问亚马逊 RDS。有关 Amazon RDS 的更多信息，请参阅 A mazon Relational Dervice

Note

以下一些示例使用 `Aws::RDS::Resource` 类的 `2.2.18` 版本中引入的方法。要运行这些示例，您必须使用该版本或更高版本的 `aws-sdk` gem。

主题

- 获取有关所有 Amazon RDS 实例的信息 (p. 120)
- 获取有关所有 Amazon RDS 快照的信息 (p. 121)
- 获取有关所有 Amazon RDS 集群及其快照的信息 (p. 121)
- 获取有关所有 Amazon RDS 安全组的信息 (p. 122)
- 获取有关所有 Amazon RDS 子网组的信息 (p. 122)
- 获取有关所有 Amazon RDS 参数组的信息 (p. 123)
- 创建 Amazon RDS 实例的快照 (p. 123)
- 创建亚马逊 RDS 集群的快照 (p. 124)

获取有关所有 Amazon RDS 实例的信息

以下示例列出了您在该us-west-2区域的所有 Amazon RDS 实例的名称 (ID) 和状态。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')
```

```
rds.db_instances.each do |i|
  puts "Name (ID): #{i.id}"
  puts "Status   : #{i.db_instance_status}"
  puts
end
```

获取有关所有 Amazon RDS 快照的信息

以下示例列出了您在该us-west-2区域的所有 Amazon RDS（实例）快照的名称 (ID) 和状态。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_snapshots.each do |s|
  puts "Name (ID): #{s.snapshot_id}"
  puts "Status:    #{s.status}"
end
```

获取有关所有 Amazon RDS 集群及其快照的信息

以下示例列出了您的所有 Amazon RDS 集群的名称 (ID) 和状态以及它们us-west-2在该区域中快照的名称 (ID) 和状态。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_clusters.each do |c|
  puts "Name (ID): #{c.id}"
  puts "Status:    #{c.status}"

  c.snapshots.each do |s|
    puts "  Snapshot: #{s.snapshot_id}"
    puts "  Status:   #{s.status}"
  end
end
```

获取有关所有 Amazon RDS 安全组的信息

以下示例列出您us-west-2在该区域中的所有 Amazon RDS 安全组的名称。

> Note
>
> Amazon RDS 安全组仅在您使用 Amazon EC2-Classic 平台时适用。如果您使用 Amazon EC2-VPC，使用 VPC 安全组。在示例中演示了这两种情况。
>
> Warning
>
> 我们将于 2022 年 8 月 15 日停用 EC2-Classic。我们建议您从 EC2-Classic 迁移到 VPC。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南或适用于 Windows 实例的 Amazon EC2 用户指南中的从 EC2-Classic 迁移到 VPC。另请参阅 EC2-Classic Networking is Retiring – Here's How to Prepare 博客文章。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_instances.each do |i|
  # Show any security group IDs and descriptions
  puts 'Security Groups:'

  i.db_security_groups.each do |sg|
    puts sg.db_security_group_name
    puts '  ' + sg.db_security_group_description
    puts
  end

  # Show any VPC security group IDs and their status
  puts 'VPC Security Groups:'

  i.vpc_security_groups.each do |vsg|
    puts vsg.vpc_security_group_id
    puts '  ' + vsg.status
    puts
  end
end
```

获取有关所有 Amazon RDS 子网组的信息

以下示例列出了该us-west-2区域内所有 Amazon RDS 子网组的名称和状态。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
```

```
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_subnet_groups.each do |s|
  puts s.name
  puts '  ' + s.subnet_group_status
end
```

获取有关所有 Amazon RDS 参数组的信息

以下示例列出该us-west-2区域内所有 Amazon RDS 参数组的名称和描述。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

rds.db_parameter_groups.each do |p|
  puts p.db_parameter_group_name
  puts '  ' + p.description
end
```

创建 Amazon RDS 实例的快照

以下示例为该区域中由 instance_nam e 表示的 Amazon RDS 实例创建快照。us-west-2

> **Note**
>
> 如果您的实例是集群的成员，则无法创建实例的快照。您必须创建集群的快照（请参阅创建 Amazon RDS 集群的快照 (p. 124)）。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'
```

```
rds = Aws::RDS::Resource.new(region: 'us-west-2')

instance = rds.db_instance(instance_name)

date =  Time.new
date_time =  date.year.to_s +  '-' +  date.month.to_s +  '-' +  date.day.to_s +  '-' +
 date.hour.to_s +  '-' +  date.min.to_s

id = instance_name + '-' + date_time

instance.create_snapshot({db_snapshot_identifier: id})

puts "Created snapshot #{id}"
```

创建亚马逊 RDS 集群的快照

以下示例为该区域中由 cluster_nam e 表示的 Amazon RDS 集群创建快照。us-west-2

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-rds'  # v2: require 'aws-sdk'

rds = Aws::RDS::Resource.new(region: 'us-west-2')

cluster = rds.db_cluster(cluster_name)

date =  Time.new
date_time =  date.year.to_s +  '-' +  date.month.to_s +  '-' +  date.day.to_s +  '-' +
 date.hour.to_s +  '-' +  date.min.to_s

id = cluster_name + '-' + date_time

cluster.create_snapshot({db_cluster_snapshot_identifier: id})

puts "Created cluster snapshot #{id}"
```

使用适用于 Ruby 的AWS开发工具包的Amazon S3 示例

Amazon Simple Storage Service (Amazon S3) 是一种面向 Internet 的存储服务。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 Amazon S3。在 Amazon S3 文档中了解有关 Amazon S3 的更多信息

主题

- 获取有关所有 Amazon S3 存储桶的信息 (p. 125)
- 获取有关区域内所有 Amazon S3 存储段的信息 (p. 126)
- 创建和使用 Amazon S3 存储桶 (p. 127)
- 确定 Amazon S3 存储桶是否存在 (p. 131)

- 获取有关 Amazon S3 存储桶对象的信息 (p. 132)
- 将对象上传到 Amazon S3 存储桶 (p. 133)
- 将包含元数据的对象上传到 Amazon S3 存储桶 (p. 134)
- 将对象从 Amazon S3 存储桶下载到文件中 (p. 135)
- 更改 Amazon S3 存储桶对象的属性 (p. 136)
- 加密 Amazon S3 存储桶项目 (p. 137)
- 将项目添加到 Amazon S3 存储桶时触发通知 (p. 150)
- 为 Amazon S3 存储桶创建LifeCycle规则配置模板 (p. 152)
- 使用 Amazon S3 存储桶策略 (p. 154)
- 为 CORS 配置Amazon S3 存储桶 (p. 158)
- 管理 Amazon S3 存储桶和对象访问权限 (p. 160)
- 使用 Amazon S3 存储桶托管网站 (p. 163)

获取有关所有 Amazon S3 存储桶的信息

以下示例列出了指定的最多 50 个可用 Amazon S3 存储桶的名称AWS 区域。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Lists the Amazon Simple Storage Service (Amazon S3) buckets owned by the
#   authenticated sender of the request.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param max_buckets [Integer] The maximum number of buckets to list. The
#   number must be between 1 and 50. If not specified, only up to the
#   first 50 objects will be listed.
# @example
#   list_buckets(Aws::S3::Client.new(region: 'us-east-1'), 25)
def list_buckets(s3_client, max_buckets = 50)
  if max_buckets < 1 || max_buckets > 50
    puts 'Maximum number of buckets to request must be between 1 and 50.'
    return
  end
  buckets = s3_client.list_buckets.buckets
  if buckets.count.zero?
    puts 'No buckets.'
    return
  else
    if buckets.count > max_buckets
      puts "First #{max_buckets} buckets:"
      i = 0
      max_buckets.times do
        puts "#{i + 1}) #{buckets[i].name}"
        i += 1
      end
    else
      puts "#{buckets.count} buckets:"
      i = 0
      buckets.count.times do
        puts "#{i + 1}) #{buckets[i].name}"
        i += 1
      end
    end
  end
rescue StandardError => e
```

```
  puts "Error listing buckets: #{e.message}"
end

def run_me
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)
  list_buckets(s3_client, 25)
end

run_me if $PROGRAM_NAME == __FILE__
```

获取有关区域内所有 Amazon S3 存储段的信息

以下代码示例列出了指定可用的 Amazon S3 存储桶的名称AWS 区域。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Checks to see which Amazon Simple Storage Service (Amazon S3)
#   buckets accessible to you and were initially created with the target
#   AWS Region specified.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param region [String] The Region to check.
# @example
#   list_accessible_buckets_in_region(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'us-east-1'
#   )
def list_accessible_buckets_in_region(s3_client, region)
  buckets = s3_client.list_buckets.buckets
  buckets_in_region = []
  buckets.each do |bucket|
    bucket_region = s3_client.get_bucket_location(
      bucket: bucket.name
    ).location_constraint
    if bucket_region == region
      buckets_in_region << bucket.name
    end
  end
  if buckets_in_region.count.zero?
    puts "No buckets accessible to you and also set to region '#{region}' " \
      'when initially created.'
    exit 1
  else
    puts "Buckets accessible to you and also set to region '#{region}' " \
      'when initially created:'
    buckets_in_region.each do |bucket_name|
      puts bucket_name
    end
  end
rescue StandardError => e
  puts "Error getting information about buckets: #{e.message}"
  exit 1
end

# Full example call:
def run_me
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  list_accessible_buckets_in_region(s3_client, region)
```

```
end

run_me if $PROGRAM_NAME == __FILE__
```

创建和使用 Amazon S3 存储桶

以下代码示例：

1. 列出 Amazon S3 中指定存储桶的可用存储桶AWS 区域。
2. 创建两个存储桶。
3. 将对象上传到其中一个存储桶。
4. 将上传的对象复制到另一个存储桶。
5. 从第一个存储桶中删除对象。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Demonstrates various Amazon S3 operations, such as creating and listing
#   buckets and uploading, copying, and deleting objects from buckets.

# Lists the available Amazon S3 buckets.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @example
#   list_buckets(Aws::S3::Client.new(region: 'us-east-1'))
def list_buckets(s3_client)
  response = s3_client.list_buckets
  if response.buckets.count.zero?
    puts 'No buckets.'
  else
    response.buckets.each do |bucket|
      puts bucket.name
    end
  end
rescue StandardError => e
  puts "Error listing buckets: #{e.message}"
end

# Creates a bucket in Amazon S3.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @return [Boolean] true if the bucket was created; otherwise, false.
# @example
#   exit 1 unless bucket_created?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_created?(s3_client, bucket_name)
  response = s3_client.create_bucket(bucket: bucket_name)
  if response.location == '/' + bucket_name
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error creating bucket: #{e.message}"
  return false
end
```

```
# Uploads an object to a bucket in Amazon S3.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name of the object.
# @param object_content [String] The content to add to the object.
# @return [Boolean] true if the object was uploaded; otherwise, false.
# @example
#   exit 1 unless object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.'
#   )
def object_uploaded?(s3_client, bucket_name, object_key, object_content)
  response = s3_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: object_content
  )
  if response.etag
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Lists available objects in an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @example
#   list_objects(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def list_objects(s3_client, bucket_name)
  response = s3_client.list_objects_v2(bucket: bucket_name)
  if response.contents.count.zero?
    puts 'No objects.'
  else
    response.contents.each do |object|
      puts object.key
    end
  end
rescue StandardError => e
  puts "Error listing objects: #{e.message}"
end

# Copies an object from one Amazon S3 bucket to another.
#
# Prerequisites:
#
# - Two Amazon S3 bucket.
```

```
# - An object in the source bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param source_bucket_name [String] The name of the source bucket.
# @param source_object_key [String] The name of the object to copy.
# @param target_bucket_name [String] The name of the target bucket.
# @param target_object_key [String] The name of the copied object.
# @return [Boolean] true if the object was copied; otherwise, false.
# @example
#   exit 1 unless object_copied?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'doc-example-bucket1',
#     'my-file-1.txt'
#   )
def object_copied?(
  s3_client,
  source_bucket_name,
  source_object_key,
  target_bucket_name,
  target_object_key
)
  response = s3_client.copy_object(
    bucket: target_bucket_name,
    copy_source: "#{source_bucket_name}/#{source_object_key}",
    key: target_object_key
  )
  if response.copy_object_result.etag
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error copying object: #{e.message}"
  return false
end

# Deletes an object from an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An object to be deleted from the bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @return object_key [String] The name of the object.
# @example
#   exit 1 unless object_deleted?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt'
#   )
def object_deleted?(s3_client, bucket_name, object_key)
  response = s3_client.delete_objects(
    bucket: bucket_name,
    delete: {
      objects: [
        {
          key: object_key
        }
      ]
    }
  )
  if response.deleted.count == 1
```

```
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error deleting object: #{e.message}"
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  object_content = 'This is the content of my-file.txt.'
  target_bucket_name = 'doc-example-bucket1'
  target_object_key = 'my-file-1.txt'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  puts 'Available buckets:'
  list_buckets(s3_client)

  if bucket_created?(s3_client, bucket_name)
    puts "Bucket '#{bucket_name}' created."
  else
    puts "Bucket '#{bucket_name}' not created. Program will stop."
    exit 1
  end

  if bucket_created?(s3_client, target_bucket_name)
    puts "Bucket '#{target_bucket_name}' created."
  else
    puts "Bucket '#{target_bucket_name}' not created. Program will stop."
    exit 1
  end

  if object_uploaded?(s3_client, bucket_name, object_key, object_content)
    puts "Object '#{object_key}' uploaded to bucket '#{bucket_name}'."
  else
    puts "Object '#{object_key}' uploaded to bucket '#{bucket_name}'. " \
      'Program will stop.'
    exit 1
  end

  puts "Objects in bucket '#{bucket_name}':"
  list_objects(s3_client, bucket_name)

  if object_copied?(
    s3_client,
    bucket_name,
    object_key,
    target_bucket_name,
    target_object_key
  )
    puts "Object '#{object_key}' copied to bucket '#{target_bucket_name}' " \
      "as object '#{target_object_key}'."
  else
    puts "Object '#{object_key}' not copied to bucket '#{target_bucket_name}' " \
    "as object '#{target_object_key}'."
  end

  if object_deleted?(s3_client, bucket_name, object_key)
    puts "Object '#{object_key}' deleted from bucket '#{bucket_name}'."
  else
    puts "Object '#{object_key}' not deleted from bucket '#{bucket_name}'."
  end
```

```
end

run_me if $PROGRAM_NAME == __FILE__
```

确定 Amazon S3 存储桶是否存在

以下代码示例检查指定的存储桶是否存在于 Amazon S3 中以及您是否可以访问。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Checks to see whether an Amazon Simple Storage Service
#   (Amazon S3) bucket exists.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket.
# @return [Boolean] true if the bucket exists; otherwise, false.
# @example
#   exit 1 unless bucket_exists?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_exists?(s3_client, bucket_name)
  response = s3_client.list_buckets
  response.buckets.each do |bucket|
    return true if bucket.name == bucket_name
  end
  return false
rescue StandardError => e
  puts "Error listing buckets: #{e.message}"
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if bucket_exists?(s3_client, bucket_name)
    puts 'Bucket exists.'
  else
    puts 'Bucket does not exist or is not accessible to you.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

以下代码示例检查指定存储桶是否存在于 Amazon S3 中以及您是否有权访问该存储桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-s3'

# Determines whether an Amazon Simple Storage Service (Amazon S3)
#   bucket exists and you have permission to access it.
#
# Prerequisites:
#
# - An S3 bucket.
```

```
#
# @param
# @param
# @return [Boolean] true if the bucket exists and you have permission to
#   access it; otherwise, false.
# @example
#   exit 1 unless bucket_exists_and_accessible?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_exists_and_accessible?(s3_client, bucket_name)
  s3_client.head_bucket(bucket: bucket_name)
  return true
rescue StandardError
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if bucket_exists_and_accessible?(s3_client, bucket_name)
    puts "Bucket '#{bucket_name}' exists and is accessible to you."
  else
    puts "Bucket '#{bucket_name}' does not exist " \
      'or is not accessible to you.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

获取有关 Amazon S3 存储桶对象的信息

以下代码示例最多列出了 Amazon S3 中指定存储桶中的前 1,000 个对象。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Lists the objects in an Amazon Simple Storage Service (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket, preferrably containing at least one object.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket.
# @param max_objects [Integer] The maximum number of objects to list. The
#   number must be between 1 and 1,000. If not specified, only up to the
#   first 50 objects will be listed.
# @example
#   list_bucket_objects(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     100
#   )
def list_bucket_objects(s3_client, bucket_name, max_objects = 50)
  if max_objects < 1 || max_objects > 1000
    puts 'Maximum number of objects to request must be between 1 and 1,000.'
    return
  end
```

```
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: max_objects
  ).contents

  if objects.count.zero?
    puts "No objects in bucket '#{bucket_name}'."
    return
  else
    if objects.count == max_objects
      puts "First #{objects.count} objects in bucket '#{bucket_name}':"
    else
      puts "Objects in bucket '#{bucket_name}':"
    end
    objects.each do |object|
      puts object.key
    end
  end
rescue StandardError => e
  puts "Error accessing bucket '#{bucket_name}' " \
    "or listing its objects: #{e.message}"
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  list_bucket_objects(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

将对象上传到 Amazon S3 存储桶

以下示例将对象上传到 Amazon S3 中的存储桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Uploads an object to a bucket in Amazon Simple Storage Service (Amazon S3).
#
# Prerequisites:
#
# - An S3 bucket.
# - An object to upload to the bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name of the object.
# @return [Boolean] true if the object was uploaded; otherwise, false.
# @example
#   exit 1 unless object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt'
#   )
def object_uploaded?(s3_client, bucket_name, object_key)
  response = s3_client.put_object(
    bucket: bucket_name,
```

```
    key: object_key
  )
  if response.etag
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if object_uploaded?(s3_client, bucket_name, object_key)
    puts "Object '#{object_key}' uploaded to bucket '#{bucket_name}'."
  else
    puts "Object '#{object_key}' not uploaded to bucket '#{bucket_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

将包含元数据的对象上传到 Amazon S3 存储桶

以下代码示例在 Amazon S3 中上传对象，将指定的元数据与上传的对象相关联。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Uploads an object to a bucket in Amazon Simple Storage Service (Amazon S3).
#   Also associates specified metadata with the uploaded object.
#
# Prerequisites:
#
# - An S3 bucket.
# - An object to upload to the bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name of the object.
# @param metadata [Hash] Metadata to associate with the uploaded object.
# @return [Boolean] true if the object was uploaded; otherwise, false.
# @example
#   exit 1 unless object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt'
#   )
def object_uploaded_with_metadata?(
  s3_client,
  bucket_name,
  object_key,
  metadata
)
  response = s3_client.put_object(
    bucket: bucket_name,
```

```
    key: object_key,
    metadata: metadata
  )
  if response.etag
    return true
  else
    return false
  end
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  metadata = {
    author: 'Mary Doe',
    version: '1.0.0.0'
  }
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if object_uploaded_with_metadata?(
    s3_client,
    bucket_name,
    object_key,
    metadata
  )
    puts "Object '#{object_key}' uploaded to bucket '#{bucket_name}' " \
      'with metadata.'
  else
    puts "Object '#{object_key}' not uploaded to bucket '#{bucket_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

将对象从 Amazon S3 存储桶下载到文件中

以下示例从 Amazon S3 存储桶下载对象。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Downloads an object from an Amazon Simple Storage Service (Amazon S3) bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket containing the object.
# @param object_key [String] The name of the object to download.
# @param local_path [String] The path on your local computer to download
#   the object.
# @return [Boolean] true if the object was downloaded; otherwise, false.
# @example
#   exit 1 unless object_downloaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     './my-file.txt'
#   )
def object_downloaded?(s3_client, bucket_name, object_key, local_path)
```

```
  s3_client.get_object(
    response_target: local_path,
    bucket: bucket_name,
    key: object_key
  )
rescue StandardError => e
  puts "Error getting object: #{e.message}"
end

def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  local_path = "./#{object_key}"
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if object_downloaded?(s3_client, bucket_name, object_key, local_path)
    puts "Object '#{object_key}' in bucket '#{bucket_name}' " \
      "downloaded to '#{local_path}'."
  else
    puts "Object '#{object_key}' in bucket '#{bucket_name}' not downloaded."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

更改 Amazon S3 存储桶对象的属性

以下代码示例将对象从一个 Amazon S3 存储桶复制到另一个。它还会为复制的对象设置预先确定的访问控制列表 (ACL) 和 Amazon S3 存储类。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-s3'

# Copies an object from one Amazon Simple Storage Service (Amazon S3)
#   bucket to another. You can also set an access control list
#   (ACL) and an S3 storage class on the copied object.
#
# Prerequisites:
#
# - A source S3 bucket and a target S3 bucket.
# - An object in the source bucket to copy.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param source_object_path [String] The path and file name of the
#   object to copy.
# @param target_bucket_name [String] The name of the destination bucket.
# @param target_object_path [String] The path and file name of the
#   copied object in the destination bucket.
# @param canned_acl [String] A predetermined ACL. Allowed values include
#   'private', 'public-read', 'public-read-write', 'authenticated-read',
#   'aws-exec-read', 'bucket-owner-read', and 'bucket-owner-full-control'.
#   If not specified, the default is 'private'.
# @param storage_class [String] The S3 storage class for the copied object.
#   Allowed values include 'STANDARD', 'REDUCED_REDUNDANCY', 'STANDARD_IA',
#   'ONEZONE_IA', 'INTELLIGENT_TIERING', 'GLACIER', and 'DEEP_ARCHIVE'.
#   If not specified, the default is 'STANDARD'.
# @return [Boolean] true if the object was copied; otherwise, false.
# @example
#   exit 1 unless object_copied_with_additional_properties?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket/my-file.txt',
```

```
#     'doc-example-bucket1',
#     'copied-files/my-copied-file.txt',
#     'bucket-owner-read',
#     'STANDARD_IA'
#   )
def object_copied_with_additional_properties?(
  s3_client,
  source_object_path,
  target_bucket_name,
  target_object_path,
  canned_acl = 'private',
  storage_class = 'STANDARD'
)
  s3_client.copy_object(
    bucket: target_bucket_name,
    copy_source: source_object_path,
    key: target_object_path,
    acl: canned_acl,
    storage_class: storage_class
  )
  return true
rescue StandardError => e
  puts "Error copying object: #{e.message}"
  return false
end

# Full example call:
def run_me
  source_object_path = 'doc-example-bucket/my-file.txt'
  target_bucket_name = 'doc-example-bucket1'
  target_object_path = 'copied-files/my-copied-file.txt'
  canned_acl = 'bucket-owner-read'
  storage_class = 'STANDARD_IA'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if object_copied_with_additional_properties?(
    s3_client,
    source_object_path,
    target_bucket_name,
    target_object_path,
    canned_acl,
    storage_class
  )
    puts "Object copied from '#{source_object_path}' to " \
      "'#{target_bucket_name}/#{target_object_path}'."
  else
    puts "Object '#{source_object_path}' not copied to " \
      "'#{target_bucket_name}/#{target_object_path}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

加密 Amazon S3 存储桶项目

Amazon S3 支持在客户端和服务器上对 Amazon S3 存储桶对象进行加密。要加密客户端上的对象，您可以自己执行加密，要么使用自己创建的密钥，要么使用 AWS Key Management Service (AWS KMS) 为您管理的密钥。

要在服务器上对对象进行加密，您有更多选项。

- 您可以让 Amazon S3 在您将对象上传到存储桶后自动对这些对象进行加密。在您为存储桶配置了此选项后，将对您此后上传的每个对象进行加密。

- 您可以让 Amazon S3 在您将对象上传到存储桶后对该对象进行加密。此方法的缺点是，您仍然可以上传未加密的对象。
- 您可以让 Amazon S3 在您将对象上传到存储桶后对该对象进行加密。此方法的缺点是，您仍然可以上传未加密的对象。

以下示例介绍了这些选项，从指定自动对上传到存储桶的所有对象进行加密的最简单示例，到在客户端上使用非对称公有和私有密钥的最复杂示例。无需担心，我们将逐步说明这些概念。在 Amazon S3 中了解有关加密的信息，请参阅使用加密保护数据。

主题

- 服务器端加密 (p. 138)
- 客户端加密 (p. 143)

服务器端加密

要在服务器上对对象进行加密，您具有以下选项。

- 您可以让 Amazon S3 在您将对象上传到存储桶后自动对这些对象进行加密。在您为存储桶配置了此选项后，将对您此后上传的每个对象进行加密。
- 您可以让 Amazon S3 在您将对象上传到存储桶后对该对象进行加密。此方法的缺点是，您仍然可以上传未加密的对象。
- 在您尝试将对象上传到存储桶时，可以让 Amazon S3 拒绝未加密的对象。

有关 Amazon S3 中的服务端加密，请参阅使用服务器端加密保护数据。

主题

- 为 Amazon S3 存储桶设置默认服务器端加密 (p. 138)
- 在服务器上对 Amazon S3 存储桶对象进行加密 (p. 139)
- 要求服务器上的加密以上传 Amazon S3 存储桶对象 (p. 140)
- 使用密钥加密 Amazon S3 存储桶对象 AWS KMS (p. 142)

为 Amazon S3 存储桶设置默认服务器端加密

以下代码示例使用 Amazon S3 存储桶设置 Amazon S3 存储桶的默认加密状态，使用服务器端加密 (S AWS KMS key E) 和。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Sets the default encryption state for an Amazon S3 bucket using
#   server-side encryption (SSE) with an
#   AWS KMS key.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An AWS KMS key.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param kms_master_key_id [String] The ID of the KMS key.
# @return [Boolean] true if the default encryption state was
#   successfully set; otherwise, false.
```

```
# @example
#   exit 1 unless default_bucket_encryption_sse_cmk_set?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     '9041e78c-7a20-4db3-929e-828abEXAMPLE'
#   )
def default_bucket_encryption_sse_cmk_set?(
  s3_client,
  bucket_name,
  kms_master_key_id
)
  s3_client.put_bucket_encryption(
    bucket: bucket_name,
    server_side_encryption_configuration: {
      rules: [
        {
          apply_server_side_encryption_by_default: {
            sse_algorithm: 'aws:kms',
            kms_master_key_id: kms_master_key_id
          }
        }
      ]
    }
  )
  return true
rescue StandardError => e
  puts "Error setting default encryption state: #{e.message}"
  return false
end

def run_me
  bucket_name = 'doc-example-bucket'
  kms_master_key_id = '9041e78c-7a20-4db3-929e-828abEXAMPLE'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if default_bucket_encryption_sse_cmk_set?(
    s3_client,
    bucket_name,
    kms_master_key_id
  )
    puts 'Default encryption state set.'
  else
    puts 'Default encryption state not set.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在服务器上对 Amazon S3 存储桶对象进行加密

以下代码示例将加密对象添加到 Amazon S3 存储桶。使用 aws/ AWS KMS key s3 在服务器上执行加密。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Adds an encrypted object to an Amazon S3 bucket. The encryption is performed
#   on the server by using the aws/s3 AWS KMS key.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
```

```
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name to assign to the uploaded object.
# @param content_to_encrypt [String] The content to be encrypted.
# @return [Boolean] true if the encrypted object was successfully uploaded;
#   otherwise, false.
# @example
#   exit 1 unless kms_sse_encrypted_object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.'
#   )
def kms_sse_encrypted_object_uploaded?(
  s3_client,
  bucket_name,
  object_key,
  content_to_encrypt
)
  s3_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: content_to_encrypt,
    server_side_encryption: 'aws:kms'
  )
  return true
rescue StandardError => e
  puts "Error uploading encrypted object: #{e.message}"
  return false
end

def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  content_to_encrypt = 'This is the content of my-file.txt.'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if kms_sse_encrypted_object_uploaded?(
    s3_client,
    bucket_name,
    object_key,
    content_to_encrypt
  )
    puts 'Encrypted object uploaded.'
  else
    puts 'Encrypted object not uploaded.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

要求服务器上的加密以上传 Amazon S3 存储桶对象

以下代码示例拒绝将未经过服务器端AWS KMS加密的对象上传到 Amazon S3 存储桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Denies uploads of objects without server-side AWS KMS encryption to
#   an Amazon S3 bucket.
```

```
#
# Prerequisites:
#
# - The Amazon S3 bucket to deny uploading objects without
#   server-side AWS KMS encryption.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if a policy was added to the bucket to
#   deny uploading objects without server-side AWS KMS encryption;
#   otherwise, false.
# @example
#   if deny_uploads_without_server_side_aws_kms_encryption?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
#     puts 'Policy added.'
#   else
#     puts 'Policy not added.'
#   end
def deny_uploads_without_server_side_aws_kms_encryption?(s3_client, bucket_name)
  policy = {
    'Version': '2012-10-17',
    'Id': 'PutObjPolicy',
    'Statement': [
      {
        'Sid': 'DenyIncorrectEncryptionHeader',
        'Effect': 'Deny',
        'Principal': '*',
        'Action': 's3:PutObject',
        'Resource': 'arn:aws:s3:::' + bucket_name + '/*',
        'Condition': {
          'StringNotEquals': {
            's3:x-amz-server-side-encryption': 'aws:kms'
          }
        }
      },
      {
        'Sid': 'DenyUnEncryptedObjectUploads',
        'Effect': 'Deny',
        'Principal': '*',
        'Action': 's3:PutObject',
        'Resource': 'arn:aws:s3:::' + bucket_name + '/*',
        'Condition': {
          'Null': {
            's3:x-amz-server-side-encryption': 'true'
          }
        }
      }
    ]
  }.to_json
  s3_client.put_bucket_policy(
    bucket: bucket_name,
    policy: policy
  )
  return true
rescue StandardError => e
  puts "Error adding policy: #{e.message}"
  return false
end

# Full example call:
def run_me
  if deny_uploads_without_server_side_aws_kms_encryption?(
    Aws::S3::Client.new(region: 'us-east-1'),
    'doc-example-bucket'
```

```
  )
    puts 'Policy added.'
  else
    puts 'Policy not added.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用密钥加密 Amazon S3 存储桶对象 AWS KMS

以下代码示例将加密对象添加到 Amazon S3。使用指定的加密密钥在服务器上执行加密。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'
require 'digest/md5'

# Adds an encrypted object to an Amazon S3 bucket. The encryption is performed
#   on the server by using the specified encryption key.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An AES256-GCM key.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name to assign to the uploaded object.
# @param content_to_encrypt [String] The content to be encrypted.
# @param encryption_key [String] The decoded representation of the
#   base64-encoded encryption key string to be used for encryption.
# @return [Boolean] true if the encrypted object was successfully uploaded;
#   otherwise, false.
# @example
#   exit 1 unless customer_key_sse_encrypted_object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.',
#     get_random_aes_256_gcm_key # See later in this file.
#   )
def customer_key_sse_encrypted_object_uploaded?(
  s3_client,
  bucket_name,
  object_key,
  content_to_encrypt,
  encryption_key
)
  s3_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: content_to_encrypt,
    sse_customer_algorithm: 'AES256',
    sse_customer_key: encryption_key,
    sse_customer_key_md5: Digest::MD5.digest(encryption_key)
  )
  return true
rescue StandardError => e
  puts "Error uploading encrypted object: #{e.message}"
  return false
end

# Generates a random AES256-GCM key. Call this function if you do not
```

```
#   already have an AES256-GCM key that you want to use to encrypt the
#   object.
#
# @ return [String] The generated AES256-GCM key. You must keep a record of
#   the key string that is reported. You will not be able to later decrypt the
#   contents of any object that is encrypted with this key unless you
#   have this key.
# @ example
#     get_random_aes_256_gcm_key
def get_random_aes_256_gcm_key
  cipher = OpenSSL::Cipher.new('aes-256-gcm')
  cipher.encrypt
  random_key = cipher.random_key
  random_key_64_string = [random_key].pack('m')
  random_key_64 = random_key_64_string.unpack('m')[0]
  puts 'The base 64-encoded string representation of the randomly-' \
    'generated AES256-GCM key is:'
  puts random_key_64_string
  puts 'Keep a record of this key string. You will not be able to later ' \
    'decrypt the contents of any object that is encrypted with this key ' \
    'unless you have this key.'
  return random_key_64
end

def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  content_to_encrypt = 'This is the content of my-file.txt.'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  # The following call generates a random AES256-GCM key. Alternatively, you can
  # provide a base64-encoded string representation of an existing key that
  # you want to use to encrypt the object. For example:#
  # encryption_key_string = 'XSiKrmzhtDKR9tTwJRSLjgwLhiMA82TC2z3GEXAMPLE='
  # encryption_key = encryption_key_string.unpack('m')[0]
  encryption_key = get_random_aes_256_gcm_key

  if customer_key_sse_encrypted_object_uploaded?(
    s3_client,
    bucket_name,
    object_key,
    content_to_encrypt,
    encryption_key
  )
    puts 'Encrypted object uploaded.'
  else
    puts 'Encrypted object not uploaded.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

客户端加密

要加密客户端上的对象，您可以自己执行加密，要么使用自己创建的密钥，要么使用 AWS Key Management Service (AWS KMS) 为您管理的密钥。

在 Amazon S3 中了解客户端加密，请参阅使用客户端加密保护数据。

主题

- 使用密钥加密 Amazon S3 存储桶对象 AWS KMS (p. 144)
- 使用密钥解密 Amazon S3 存储桶对象 AWS KMS (p. 145)

- 创建公有和私有非对称密钥 (p. 146)
- 使用公有密钥对 Amazon S3 存储桶对象进行加密 (p. 147)
- 使用私有密钥对 Amazon S3 存储桶对象进行解密 (p. 149)

使用密钥加密 Amazon S3 存储桶对象 AWS KMS

以下代码示例将加密对象上传到 Amazon S3 存储桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Uploads an encrypted object to an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An encrypted object to upload to the bucket.
#
# @param s3_encryption_client [Aws::S3::EncryptionV2::Client]
#   An initialized Amazon S3 V2 encryption client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name of the object to upload.
# @param object_content [String] The content of the object to upload.
# @return [Boolean] true if the object was encrypted and uploaded;
#   otherwise, false.
# @example
#   s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
#     region: 'us-east-1',
#     kms_key_id: '9041e78c-7a20-4db3-929e-828abEXAMPLE',
#     key_wrap_schema: :kms_context,
#     content_encryption_schema: :aes_gcm_no_padding,
#     security_profile: :v2
#   )
#   if encrypted_object_uploaded?(
#     s3_encryption_client,
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.'
#   )
#     puts 'Uploaded.'
#   else
#     puts 'Not uploaded.'
#   end
def encrypted_object_uploaded?(
  s3_encryption_client,
  bucket_name,
  object_key,
  object_content
)
  s3_encryption_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: object_content
  )
  return true
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Full example call:
```

```
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  region = 'us-east-1'
  kms_key_id = '9041e78c-7a20-4db3-929e-828abEXAMPLE'
  object_content = File.read(object_key)

  # Note that in the following call:
  # - key_wrap_schema must be kms_context for AWS KMS.
  # - To allow reading and decrypting objects that are encrypted by the
  #   Amazon S3 V1 encryption client instead, use :v2_and_legacy instead of :v2.
  s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
    region: region,
    kms_key_id: kms_key_id,
    key_wrap_schema: :kms_context,
    content_encryption_schema: :aes_gcm_no_padding,
    security_profile: :v2
  )

  if encrypted_object_uploaded?(
    s3_encryption_client,
    bucket_name,
    object_key,
    object_content
  )
    puts 'Uploaded.'
  else
    puts 'Not uploaded.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用密钥解密 Amazon S3 存储桶对象 AWS KMS

以下代码示例获取 Amazon S3 存储桶中加密对象的内容。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Gets the contents of an encrypted object in an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An encrypted object in the bucket to get.
#
# @param s3_encryption_client [Aws::S3::EncryptionV2::Client]
#   An initialized Amazon S3 V2 encryption client.
# @param bucket_name [String] The name of the bucket.
# @param object_key [String] The name of the encrypted object to get.
# @return [String] If successful, the object's content; otherwise,
#   diagnostic information about the unsuccessful attempt.
# @example
#   s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
#     region: 'us-east-1',
#     kms_key_id: '9041e78c-7a20-4db3-929e-828abEXAMPLE',
#     key_wrap_schema: :kms_context,
#     content_encryption_schema: :aes_gcm_no_padding,
#     security_profile: :v2
#   )
#   puts get_decrypted_object_content(
```

```
#     s3_encryption_client,
#     'doc-example-bucket',
#     'my-file.txt'
#   )
def get_decrypted_object_content(
  s3_encryption_client,
  bucket_name,
  object_key
)
  response = s3_encryption_client.get_object(
    bucket: bucket_name,
    key: object_key
  )
  if defined?(response.body)
    return response.body.read
  else
    return 'Error: Object content empty or unavailable.'
  end
rescue StandardError => e
  return "Error getting object content: #{e.message}"
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  region = 'us-east-1'
  kms_key_id = '9041e78c-7a20-4db3-929e-828abEXAMPLE'

  # Note that in the following call:
  # - key_wrap_schema must be kms_context for AWS KMS.
  # - To allow reading and decrypting objects that are encrypted by the
  #   Amazon S3 V1 encryption client instead, use :v2_and_legacy instead of :v2.
  s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
    region: region,
    kms_key_id: kms_key_id,
    key_wrap_schema: :kms_context,
    content_encryption_schema: :aes_gcm_no_padding,
    security_profile: :v2
  )

  puts get_decrypted_object_content(
    s3_encryption_client,
    bucket_name,
    object_key
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

请参阅上的完整示例GitHub。

创建公有和私有非对称密钥

以下代码示例创建公有密钥和私有密钥文件对。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'openssl'

# Creates a public and private key file pair.
#
# @param public_key_file [File] An instance of the public key file.
```

```
# @param private_key_file [File] An instance of the private key file.
# @param passphrase [String] A passphrase for the private key file.
# @return [Boolean] true if the public and private key files were created;
#   otherwise, false.
# @example
#   public_key_file = File.new('public_key.pem', 'w')
#   private_key_file = File.new('private_key.pem', 'w')
#   exit 1 unless public_and_private_key_created?(
#     public_key_file,
#     private_key_file,
#     'my-passphrase'
#   )
def public_and_private_key_created?(
  public_key_file,
  private_key_file,
  passphrase
)

  key = OpenSSL::PKey::RSA.new(2048)

  public_key_file.write(key.public_key.to_pem)
  public_key_file.close

  cipher = OpenSSL::Cipher.new('AES-128-CBC')
  key_secure = key.export(cipher, passphrase)

  private_key_file.write(key_secure)
  private_key_file.close

  return true
rescue StandardError => e
  puts 'Could not create the public key file, the private key file, ' \
    "or both: #{e.message}"
end

# Full example call:
def run_me
  public_key_file_name = 'public_key.pem'
  public_key_file = File.new(public_key_file_name, 'w')
  private_key_file_name = 'private_key.pem'
  private_key_file = File.new(private_key_file_name, 'w')
  passphrase = 'my-passphrase'

  puts "Creating public key file at '#{public_key_file_name}', and " \
    "creating private key file at '#{private_key_file_name}' with passphrase " \
    "'#{passphrase}'..."

  if public_and_private_key_created?(
    public_key_file,
    private_key_file,
    passphrase)
    puts 'Public and private key file pair created.'
  else
    exit 1
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用公有密钥对 Amazon S3 存储桶对象进行加密

以下代码示例将对象上传到 Amazon S3 存储桶。该对象的内容使用 RSA 公钥加密。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'
require 'openssl'

# Uploads an object to an Amazon S3 bucket. The object's contents
#   are encrypted with an RSA public key.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_encryption_client [Aws::S3::EncryptionV2::Client] An initialized
#   Amazon S3 encryption client.
# @param bucket_name [String] The bucket's name.
# @param object_key [String] The name of the object.
# @param object_content [String] The content to add to the object.
# @return [Boolean] true if the object was uploaded; otherwise, false.
# @example
#   exit 1 unless object_uploaded_with_public_key_encryption?(
#     Aws::S3::EncryptionV2::Client.new(
#       encryption_key: OpenSSL::PKey::RSA.new(File.read('my-public-key.pem')),
#       key_wrap_schema: :rsa_oaep_sha1,
#       content_encryption_schema: :aes_gcm_no_padding,
#       security_profile: :v2,
#       region: 'us-east-1'
#     ),
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.'
#   )
def object_uploaded_with_public_key_encryption?(
  s3_encryption_client,
  bucket_name,
  object_key,
  object_content
)
  s3_encryption_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: object_content
  )
  return true
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Full example call:
# Prerequisites: an RSA key pair.
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  object_content = 'This is the content of my-file.txt.'
  region = 'us-east-1'
  public_key_file = 'my-public-key.pem'
  public_key = OpenSSL::PKey::RSA.new(File.read(public_key_file))

  # When initializing this Amazon S3 encryption client, note:
  # - For key_wrap_schema, use rsa_oaep_sha1 for asymmetric keys.
  # - For security_profile, for reading or decrypting objects encrypted
  #     by the v1 encryption client, use :v2_and_legacy instead.
  s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
    encryption_key: public_key,
    key_wrap_schema: :rsa_oaep_sha1,
    content_encryption_schema: :aes_gcm_no_padding,
```

```
    security_profile: :v2,
    region: region
  )

  if object_uploaded_with_public_key_encryption?(
    s3_encryption_client,
    bucket_name,
    object_key,
    object_content
  )
    puts 'Object uploaded.'
  else
    puts 'Object not uploaded.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用私有密钥对 Amazon S3 存储桶对象进行解密

以下代码示例从 Amazon S3 存储桶下载数据元。该对象的内容最初是使用 RSA 公钥加密的。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'
require 'openssl'

# Downloads an object from an Amazon S3 bucket. The object's contents
#   were originally encrypted with an RSA public key.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An object in this bucket.
#
# @param s3_encryption_client [Aws::S3::EncryptionV2::Client] An initialized
#   Amazon S3 encryption client.
# @param bucket_name [String] The bucket's name.
# @param object_key [String] The name of the object.
# @return [String] The object's content; otherwise, information about the
#   failed download operation.
# @example
#   puts download_object_with_private_key_encryption(
#     Aws::S3::EncryptionV2::Client.new(
#       encryption_key: OpenSSL::PKey::RSA.new(File.read('my-private-key.pem')),
#       key_wrap_schema: :rsa_oaep_sha1,
#       content_encryption_schema: :aes_gcm_no_padding,
#       security_profile: :v2,
#       region: 'us-east-1'
#     ),
#     'doc-example-bucket',
#     'my-file.txt'
#   )
def download_object_with_private_key_encryption(
  s3_encryption_client,
  bucket_name,
  object_key
)
  response = s3_encryption_client.get_object(
    bucket: bucket_name,
    key: object_key
  )
  return response.body.read
```

```
rescue StandardError => e
  puts "Error downloading object: #{e.message}"
end

# Full example call:
# Prerequisites: the same RSA key pair you originally used to encrypt the object.
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  region = 'us-east-1'
  private_key_file = 'my-private-key.pem'
  private_key = OpenSSL::PKey::RSA.new(File.read(private_key_file))

  # When initializing this Amazon S3 encryption client, note:
  # - For key_wrap_schema, use rsa_oaep_sha1 for asymmetric keys.
  # - For security_profile, for reading or decrypting objects encrypted
  #     by the v1 encryption client, use :v2_and_legacy instead.
  s3_encryption_client = Aws::S3::EncryptionV2::Client.new(
    encryption_key: private_key,
    key_wrap_schema: :rsa_oaep_sha1,
    content_encryption_schema: :aes_gcm_no_padding,
    security_profile: :v2,
    region: region
  )
  puts "The content of '#{object_key}' in bucket '#{bucket_name}' is:"
  puts download_object_with_private_key_encryption(
    s3_encryption_client,
    bucket_name,
    object_key
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

将项目添加到 Amazon S3 存储桶时触发通知

以下代码示例将事件通知添加到 Amazon S3 存储桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Adds an event notification to an Amazon Simple Storage Service
#   (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket.
# - For an event notification to AWS Lambda, a Lambda function.
# - For an event notification to Amazon Simple Notification Service
#   (Amazon SNS), an SNS topic.
# - For an event notification to Amazon Simple Queue Service
#   (Amazon SQS), an SQS queue.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The name of the bucket.
# @param events [Array] The S3 events to notify on.
# @param send_to_type [String] The type of AWS resource to notify. Allowed
#   values include 'lambda' for Lambda, 'sns' for SNS, and 'sqs' for SQS.
# @param resource_arn [String] The Amazon Resource Name (ARN) of the
#   AWS resource.
# @return [Boolean] true if the bucket notification configuration was set;
#   otherwise, false.
```

```
# @example
#   exit 1 unless bucket_notification_configuration_set?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     ['s3:ObjectCreated:*'],
#     'sns',
#     'arn:aws:sns:us-east-1:111111111111:my-topic'
#   )
def bucket_notification_configuration_set?(
  s3_client,
  bucket_name,
  events,
  send_to_type,
  resource_arn
)
  case send_to_type
  when 'lambda'
    s3_client.put_bucket_notification_configuration(
      bucket: bucket_name,
      notification_configuration: {
        lambda_function_configurations: [
          {
            lambda_function_arn: resource_arn,
            events: events
          }
        ]
      }
    )
  when 'sns'
    s3_client.put_bucket_notification_configuration(
      bucket: bucket_name,
      notification_configuration: {
        topic_configurations: [
          {
            topic_arn: resource_arn,
            events: events
          }
        ]
      }
    )
  when 'sqs'
    s3_client.put_bucket_notification_configuration(
      bucket: bucket_name,
      notification_configuration: {
        queue_configurations: [
          {
            queue_arn: resource_arn,
            events: events
          }
        ]
      }
    )
  else
    puts 'Error setting bucket notification configuration: ' \
      "Cannot determine send-to type. Must be 'lambda', 'sns', or 'sqs'."
    return false
  end
  return true
rescue StandardError => e
  puts "Error setting bucket notification configuration: #{e.message}"
  return false
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
```

```
  events = ['s3:ObjectCreated:*']

  # For an SNS topic:
  send_to_type = 'sns'
  resource_arn = 'arn:aws:sns:us-east-1:111111111111:my-topic'

  # For an SQS queue:
  # send_to_type = 'sqs'
  # resource_arn = 'arn:aws:sqs:us-east-1:111111111111:my-queue'

  # For a Lambda function:
  # send_to_type = 'lambda'
  # resource_arn = 'arn:aws:lambda:us-east-1:111111111111:function:myFunction'

  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if bucket_notification_configuration_set?(
    s3_client,
    bucket_name,
    events,
    send_to_type,
    resource_arn
  )
    puts 'Bucket notification configuration set.'
  else
    puts 'Bucket notification configuration not set.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

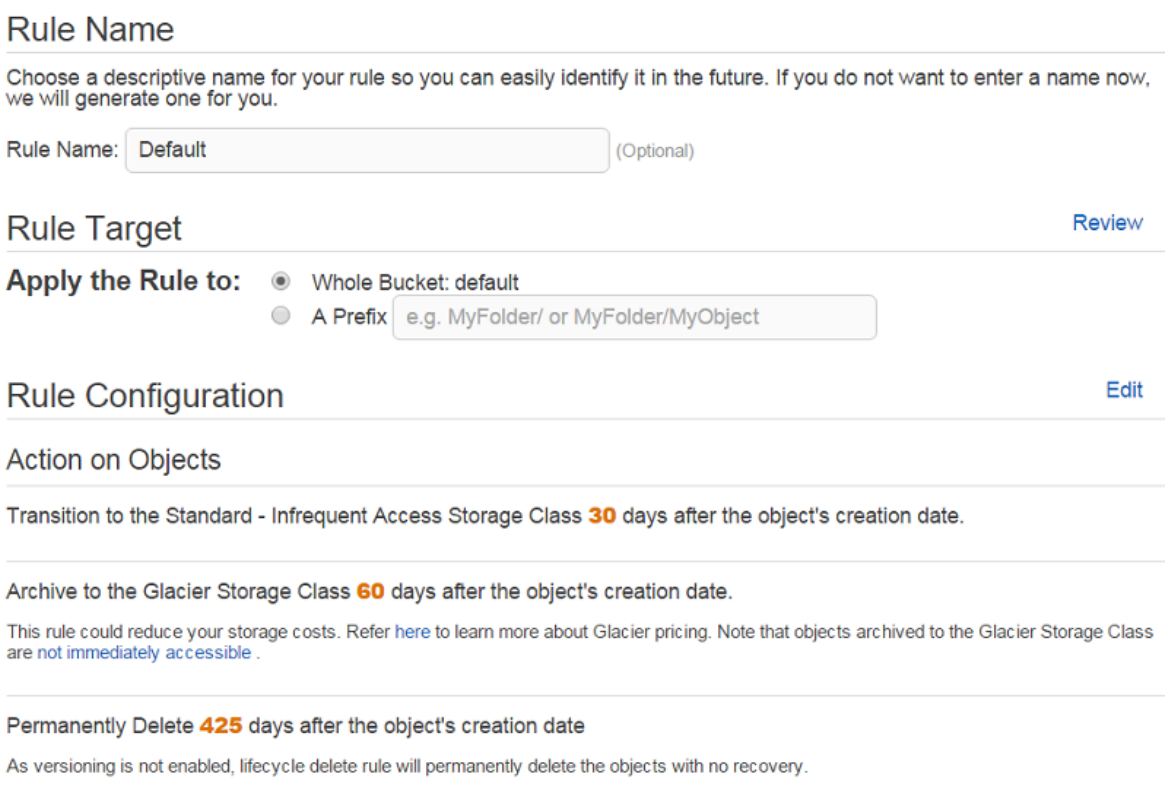
为 Amazon S3 存储桶创建LifeCycle规则配置模板

如果您有（或计划创建）数量不小的对象，并且想要指定何时将它们移至长期存储或删除它们，则可以通过为生命周期规则创建模板并将该模板应用于所有 Amazon S3 存储桶来节省大量时间。

此过程包括以下步骤：

1. 手动修改现有存储桶上的生命周期设置。
2. 保存规则。
3. 将规则应用于您的其他存储桶。

从以下规则开始：



运行以下代码来生成该规则的 JSON 表示形式。将输出另存为 `default.json`。

```
require 'aws-sdk'

s3 = Aws::S3::Client.new(region: 'us-west-2')
resp = s3.get_bucket_lifecycle_configuration(bucket: 'default')

resp.rules.each do |rule|
  rule.to_hash.to_json
end
```

输出应与以下内容类似。

```
[{"expiration":
{"date":null,"days":425},"id":"default","prefix":"","status":"Enabled","transitions":
[{"date":null,"days":30,"storage_class":"STANDARD_IA"},
{"date":null,"days":60,"storage_class":"GLACIER"}],"noncurrent_version_transitions":
[],"noncurrent_version_expiration":null}]
```

现在，您具有生命周期规则的 JSON，您可以使用以下示例将其应用于任何其他存储桶。此示例从 `default.json` 获取规则并将其应用于存储桶 `other_bucket`。

```
require 'aws-sdk'
require 'json'

class Aws::S3::Types::LifecycleExpiration
  def to_map
    map = Hash.new
    self.members.each { |m| map[m] = self[m] }
    map
  end
```

```
  def to_json(*a)
    to_map.to_json(*a)
  end
end

class Aws::S3::Types::Transition
  def to_map
    map = Hash.new
    self.members.each { |m| map[m] = self[m] }
    map
  end

  def to_json(*a)
    to_map.to_json(*a)
  end
end

class Aws::S3::Types::LifecycleRule
  def to_map
    map = Hash.new
    self.members.each { |m| map[m] = self[m] }
    map
  end

  def to_json(*a)
    to_map.to_json(*a)
  end
end

# Pull in contents as a string
value = File.open('default.json', "rb").read
json_data = JSON.parse(value, opts={symbolize_names: true})

s3 = Aws::S3::Client.new(region: 'us-west-2')
s3.put_bucket_lifecycle_configuration(:bucket => 'other_bucket', :lifecycle_configuration
 => {:rules => json_data})
```

Note

最佳实践
建议您在 Amazon S3 存储桶上启用AbortIncompleteMultipartUpload生命周期规则。
该规则指示 Amazon S3 中止未在启动后的指定天数内完成的分段上传。当超过设定的时间限制时，Amazon S3 会中止上传，然后删除不完整的上传数据。
有关更多信息，请参阅《Amazon S imple Storage Service 用户指南》中的在启用了版本控制的存储桶中进行生命周期配置。

使用 Amazon S3 存储桶策略

以下代码示例显示了如何：

1. 在 Simple Storage Service（Amazon S3）中创建一个存储桶。
2. 将存储桶策略添加到存储桶。
3. 更改存储桶策略。
4. 从存储桶移除存储桶策略。
5. 删除桶。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0
```

```
# This code example demonstrates how to:
# - Create a bucket in Amazon Simple Storage Service (Amazon S3).
# - Add a bucket policy to the bucket.
# - Change the bucket policy.
# - Remove the bucket policy from the bucket.
# - Delete the bucket.

require 'aws-sdk-s3'
require 'securerandom'

# Creates an Amazon Simple Storage Service (Amazon S3) bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the bucket was created; otherwise, false.
# @example
#   exit 1 unless bucket_created?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_created?(s3_client, bucket_name)
  s3_client.create_bucket(bucket: bucket_name)
  return true
rescue StandardError => e
  puts "Error creating bucket: #{e.message}"
  return false
end

# Adds a bucket policy to an Amazon Simple Storage Service (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket.
# - A valid AWS principal Amazon Resource Name (ARN).
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The bucket's name.
# @param aws_principal [String] The ARN of the AWS principal to allow.
# @param action [String] The bucket action to allow.
# @return [Boolean] true if the bucket policy was added; otherwise, false.
# @example
#   exit 1 unless bucket_policy_added?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'arn:aws:iam::111111111111:user/SomeUser',
#     's3:GetObject'
#   )
def bucket_policy_added?(s3_client, bucket_name, aws_principal, action)
  bucket_policy = {
    'Version' => '2012-10-17',
    'Statement' => [
      {
        'Effect' => 'Allow',
        'Principal' => {
          'AWS' => aws_principal
        },
        'Action' => action,
        'Resource' => "arn:aws:s3:::#{bucket_name}/*"
      }
    ]
  }.to_json
  s3_client.put_bucket_policy(
    bucket: bucket_name,
    policy: bucket_policy
  )
  return true
```

```
rescue StandardError => e
  puts "Error adding bucket policy: #{e.message}"
  return false
end

# Updates the AWS principal Amazon Resource Name (ARN) in an existing
#   bucket policy for an Amazon Simple Storage Service (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket.
# - A bucket policy attached to the bucket.
# - A valid AWS principal ARN.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The bucket's name.
# @param aws_principal [String] The ARN of the new AWS principal to allow.
# @return [Boolean] true if the bucket policy was updated; otherwise, false.
# @example
#   exit 1 unless bucket_policy_aws_principal_updated?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'arn:aws:iam::111111111111:user/SomeOtherUser'
#   )
def bucket_policy_aws_principal_updated?(
  s3_client,
  bucket_name,
  new_aws_principal
)
  bucket_policy = s3_client.get_bucket_policy(bucket: bucket_name).policy.read
  policy_json = JSON.parse(bucket_policy)

  policy_json['Statement'][0]['Principal']['AWS'] = new_aws_principal

  s3_client.put_bucket_policy(
    bucket: bucket_name,
    policy: policy_json.to_json
  )
  return true
rescue StandardError => e
  puts "Error updating bucket policy: #{e.message}"
  return false
end

# Deletes a bucket policy for an Amazon Simple Storage Service
#   (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the bucket policy was deleted; otherwise, false.
# @example
#   exit 1 unless bucket_policy_deleted?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_policy_deleted?(s3_client, bucket_name)
  s3_client.delete_bucket_policy(bucket: bucket_name)
  return true
rescue StandardError => e
  puts "Error deleting bucket policy: #{e.message}"
  return false
end
```

```
# Deletes an Amazon Simple Storage Service (Amazon S3) bucket.
#
# Prerequisites:
#
# - An S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the bucket was deleted; otherwise, false.
# @example
#   exit 1 unless bucket_deleted?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket'
#   )
def bucket_deleted?(s3_client, bucket_name)
  s3_client.delete_bucket(bucket: bucket_name)
  return true
rescue StandardError => e
  puts "Error deleting bucket: #{e.message}"
  return false
end

# Full example call:
def run_me
  aws_principal = 'arn:aws:iam::111111111111:user/SomeUser'
  new_aws_principal = 'arn:aws:iam::111111111111:user/SomeOtherUser'
  action = 's3:GetObject'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)
  bucket_name = 'bucket-' + SecureRandom.uuid

  if bucket_created?(s3_client, bucket_name)
    puts "Bucket '#{bucket_name}' created."
  else
    puts "Bucket '#{bucket_name}' not created. Stopping program."
    exit 1
  end

  if bucket_policy_added?(s3_client, bucket_name, aws_principal, action)
    puts 'Bucket policy added.'
  else
    puts 'Bucket policy not added.'
  end

  if bucket_policy_aws_principal_updated?(
    s3_client,
    bucket_name,
    new_aws_principal
  )
    puts 'Bucket policy updated with new AWS principal.'
  else
    puts 'Bucket policy not updated with new AWS principal.'
  end

  if bucket_policy_deleted?(s3_client, bucket_name)
    puts 'Bucket policy (if any) deleted.'
  else
    puts 'Bucket policy (if any) not deleted.'
  end

  if bucket_deleted?(s3_client, bucket_name)
    puts "Bucket '#{bucket_name}' deleted."
  else
    puts "Bucket '#{bucket_name}' not deleted. " \
      'You must delete this bucket yourself.'
```

```
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

为 CORS 配置Amazon S3 存储桶

以下代码示例向 Amazon S3 存储桶添加了一个包含单一规则的跨源资源共享 (COR) 配置。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Adds a cross-origin resource sharing (CORS) configuration containing
#   a single rule to an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param allowed_methods [Array] The types of requests to allow, such as GET.
# @param allowed_origins [Array] The origins to allow, for example
#   http://www.example.com.
# @param allowed_headers [Array] The preflight request headers to allow, for
#   example x-amz-*.
# @param expose_headers [Array] The headers in the response that you want
#   callers to be able to access from their applications, for example
#   Content-Type.
# @param max_age_seconds [Integer] The maximum number of seconds
#   that your browser can cache the response for a preflight request
#   as identified by the resource, the HTTP method, and the origin.
# @returns [Boolean] true if the CORS rule was successfully set;
#   otherwise, false.
# @example
#   exit 1 unless if bucket_cors_rule_set?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     %w[GET PUT POST DELETE],
#     %w[http://www.example.com],
#     %w[*],
#     %w[x-amz-server-side-encryption x-amz-request-id x-amz-id-2],
#     3000
#   )
def bucket_cors_rule_set?(
  s3_client,
  bucket_name,
  allowed_methods = %w[GET PUT POST DELETE HEAD],
  allowed_origins = %w[*],
  allowed_headers = nil,
  expose_headers = nil,
  max_age_seconds = nil
)
  methods = []
  if allowed_methods.count.zero?
    puts 'Error: No CORS methods provided.'
    return false
  else
    allowed_methods.each do |method|
      case method.upcase
      when 'GET', 'PUT', 'POST', 'DELETE', 'HEAD'
        methods.append(method)
```

```
      else
        puts "Error: '#{method}' is not an allowed CORS method."
        return false
      end
    end
  end
  s3_client.put_bucket_cors(
    bucket: bucket_name,
    cors_configuration: {
      cors_rules: [
        {
          allowed_headers: allowed_headers,
          allowed_methods: methods,
          allowed_origins: allowed_origins,
          expose_headers: expose_headers,
          max_age_seconds: max_age_seconds
        }
      ]
    }
  )
  return true
rescue StandardError => e
  puts "Error setting CORS methods: #{e.message}"
  return false
end

# Gets the cross-origin resource sharing (CORS) rules for an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @returns [Array<Aws::S3::Types::CORSRule>] The list of CORS rules.
# @example
#   puts bucket_cors_rules(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket')
def bucket_cors_rules(s3_client, bucket_name)
  response = s3_client.get_bucket_cors(bucket: bucket_name)
  response.cors_rules
rescue StandardError => e
  puts "Error getting CORS rules: #{e.message}"
end

def run_me
  bucket_name = 'doc-example-bucket'
  allowed_methods = %w[GET PUT POST DELETE]
  allowed_origins = %w[http://www.example.com]
  allowed_headers = %w[*]
  expose_headers = %w[x-amz-server-side-encryption x-amz-request-id x-amz-id-2]
  max_age_seconds = 3000
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if bucket_cors_rule_set?(
    s3_client,
    bucket_name,
    allowed_methods,
    allowed_origins,
    allowed_headers,
    expose_headers,
    max_age_seconds
  )
    puts 'CORS rule set. Current rules:'
```

```
    puts bucket_cors_rules(s3_client, bucket_name)
  else
    puts 'CORS rule not set.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

管理 Amazon S3 存储桶和对象访问权限

以下代码示例显示了如何：

1. 将 Amazon S3 存储桶的初始访问级别设置为私有。
2. 尝试访问对象并将其上传到存储桶，但应该会失败。
3. 将访问级别设置为公共读取。
4. 尝试访问对象并将其上传到存储桶，现在应该会成功。
5. 将访问级别设置回私有。
6. 尝试访问对象并将其上传到存储桶，现在应该会失败。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'
require 'net/http'

# Sets the access control list (ACL) for an Amazon S3 bucket
#   for public access.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @param access_level [String] The access level for the bucket. Allowed values
#   include private, public-read, public-read-write, and authenticated-read.
# @return [Boolean] true if the ACL was set; otherwise, false.
# @example
#   exit 1 unless bucket_acl_set?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'private'
#   )
def bucket_acl_set?(s3_client, bucket_name, access_level)
  s3_client.put_bucket_acl(
    bucket: bucket_name,
    acl: access_level
  )
  return true
rescue StandardError => e
  puts "Error setting bucket ACL: #{e.message}"
  return false
end

# Uploads an object to an Amazon S3 bucket.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
```

```
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @param object_key [String] The name of the object.
# @param object_content [String] The content to add to the object.
# @return [Boolean] true if the object was uploaded; otherwise, false.
# @example
#   exit 1 unless object_uploaded?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'This is the content of my-file.txt.'
#   )
def object_uploaded?(s3_client, bucket_name, object_key, object_content)
  s3_client.put_object(
    bucket: bucket_name,
    key: object_key,
    body: object_content
  )
  return true
rescue StandardError => e
  puts "Error uploading object: #{e.message}"
  return false
end

# Sets the access control list (ACL) for an object in an
#   Amazon S3 bucket for public access.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An object in the bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @param object_key [String] The name of the object.
# @param access_level [String] The access level for the bucket. Allowed values
#   include private, public-read, public-read-write, and authenticated-read.
# @return [Boolean] true if the ACL was set; otherwise, false.
# @example
#   exit 1 unless object_acl_set?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'my-file.txt',
#     'private'
#   )
def object_acl_set?(s3_client, bucket_name, object_key, access_level)
  s3_client.put_object_acl(
    bucket: bucket_name,
    key: object_key,
    acl: access_level
  )
  return true
rescue StandardError => e
  puts "Error setting object ACL: #{e.message}"
  return false
end

# Prints information about the Amazon S3 bucket at the given path.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
#
# @param bucket_name [String] The bucket's name.
# @param region [String] The AWS Region for the bucket.
# @example
```

```
#   object_content_by_bucket_unsigned_request(
#     'doc-example-bucket',
#     'us-east-1'
#   )
def object_content_by_bucket_unsigned_request(bucket_name, region)
  bucket_path = "https://s3.#{region}.amazonaws.com/#{bucket_name}"
  response = Net::HTTP.get(URI(bucket_path))
  puts "Content of unsigned request to '#{bucket_path}':\n\n#{response}\n\n"
end

# Prints information about the Amazon S3 object in the bucket
#   at the given path.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - An object in the bucket.
#
# @param bucket_name [String] The bucket's name.
# @param object_key [String] The name of the object in the bucket.
# @param region [String] The AWS Region for the bucket.
# @example
#   object_content_by_object_unsigned_request(
#     'doc-example-bucket',
#     'my-file.txt',
#     'us-east-1'
#   )
def object_content_by_object_unsigned_request(bucket_name, object_key, region)
  object_path = "https://s3.#{region}.amazonaws.com/#{bucket_name}/#{object_key}"
  response = Net::HTTP.get(URI(object_path))
  puts "Content of unsigned request to '#{object_path}':\n\n#{response}\n\n"
end

# Full example call:
def run_me
  bucket_name = 'doc-example-bucket'
  object_key = 'my-file.txt'
  object_content = 'This is the content of my-file.txt.'
  access_level_before = 'private'
  access_level_after = 'public-read'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  # Set the initial access level of the bucket to 'private'
  #   for public access.
  if bucket_acl_set?(s3_client, bucket_name, access_level_before)
    puts "1. Initial bucket ACL set to '#{access_level_before}' " \
      "for public access.\n\n"
  else
    puts "1. Initial bucket ACL not set to '#{access_level_before}' " \
      'for public access. Stopping program.'
    exit 1
  end

  # What happens when you try to access the bucket? (It should be denied.)
  puts "2. After initial bucket ACL set to '#{access_level_before}' " \
    "for public access, trying to access the bucket:\n\n"
  object_content_by_bucket_unsigned_request(bucket_name, region)

  # Upload an object to the bucket.
  if object_uploaded?(s3_client, bucket_name, object_key, object_content)
    puts "3. Object uploaded to bucket.\n\n"
  else
    puts '3. Object not uploaded to bucket. Stopping program. ' \
      "Note that the bucket ACL is still set to '#{access_level_before}' " \
      'for public access.'
```

```
    exit 1
  end

  # What happens when you try to access the object now?
  #   (It should still be denied.)
  puts "4. After object uploaded, trying to access the object:\n\n"
  object_content_by_object_unsigned_request(bucket_name, object_key, region)

  # Now set the initial access level of the object to 'public-read'
  #   for public access.
  if object_acl_set?(s3_client, bucket_name, object_key, access_level_after)
    puts "5. Object ACL set to '#{access_level_after}' for public access.\n\n"
  else
    puts "5. Object ACL not set to '#{access_level_after}' for public " \
      'access. Stopping program. ' \
      "Note that the bucket ACL is still set to '#{access_level_before}' " \
      'for public access.'
    exit 1
  end

  # What happens when you try to access the object now? (It should now work.)
  puts "6. After object ACL set to '#{access_level_after}' for public " \
    "access, trying to access the object:\n\n"
  object_content_by_object_unsigned_request(bucket_name, object_key, region)

  # Now set the access level for the object to 'private' for public access.
  if object_acl_set?(s3_client, bucket_name, object_key, access_level_before)
    puts "7. Object ACL now set to '#{access_level_before}' " \
      "for public access.\n\n"
  else
    puts "7. Object ACL not set to '#{access_level_before}' " \
      'for public access. Stopping program. ' \
      "Note that the bucket ACL is still set to '#{access_level_before}'."
    exit 1
  end

  # What happens when you try to access the object now?
  #   (It should now be denied.)
  puts "8. After object ACL set to '#{access_level_before}' " \
    "for public access, trying to access the object:\n\n"
  object_content_by_object_unsigned_request(bucket_name, object_key, region)

  puts '9. Program ends. Note that the bucket ACL is still set to ' \
    "'#{access_level_before}' for public access."
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 Amazon S3 存储桶托管网站

以下代码示例显示如何配置 Amazon S3 存储桶作为静态网站。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX - License - Identifier: Apache - 2.0

require 'aws-sdk-s3'

# Configures an Amazon S3 bucket as a static website.
#
# Prerequisites:
#
# - An Amazon S3 bucket.
# - A file in the bucket representing the website's home or
#     default page.
```

```
# - A file in the bucket representing the website's error page.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @param index_document [String] The file name of the home or default page
#   of the website.
# @param error_document [String] The file name of the page returned when a
#   website error occurs.
# @return [Boolean] true if the bucket was successfully configured;
#   otherwise, false.
# @example
#   exit 1 unless bucket_website_configured?(
#     Aws::S3::Client.new(region: 'us-east-1'),
#     'doc-example-bucket',
#     'index.html',
#     '404.html'
#   )
def bucket_website_configured?(
  s3_client,
  bucket_name,
  index_document,
  error_document
)
  s3_client.put_bucket_website(
    bucket: bucket_name,
    website_configuration: {
      index_document: {
        suffix: index_document
      },
      error_document: {
        key: error_document
      }
    }
  )
  return true
rescue StandardError => e
  puts "Error configuring bucket as a static website: #{e.message}"
  return false
end

def run_me
  bucket_name = 'doc-example-bucket'
  index_document = 'index.html'
  error_document = '404.html'
  region = 'us-east-1'
  s3_client = Aws::S3::Client.new(region: region)

  if bucket_website_configured?(
    s3_client,
    bucket_name,
    index_document,
    error_document
  )
    puts 'Bucket configured as a static website.'
  else
    puts 'Bucket not configured as a static website.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

使用 Ruby AWS 开发工具包的Amazon SES 示例

Amazon Simple Email Service (Amazon SES) 是一个易于使用且经济高效的电子邮件平台，有便于您通过该平台，使用您自己的电子邮件地址和域来发送或接收电子邮件。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 Amazon SES。有关 Amazon SES 的更多信息，请参阅 Amazon SES 文档。

主题

- 列出有效的Amazon SES 电子邮件地址 (p. 165)
- 在Amazon SES 中验证电子邮件地址 (p. 165)
- 向 Amazon SES 中的电子邮件地址发送消息 (p. 166)
- 获取Amazon SES 统计数据 (p. 167)

列出有效的Amazon SES 电子邮件地址

以下示例演示如何使用适用于 Ruby 的 SES AWS 开发工具包列出有效的 Amazon SES 电子邮件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses'  # v2: require 'aws-sdk'

# Create client in us-west-2 region
client = Aws::SES::Client.new(region: 'us-west-2')

# Get up to 1000 identities
ids = client.list_identities({
  identity_type: "EmailAddress"
})

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
    identities: [email]
  })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  if status == "Success"
    puts email
  end
end
```

请参阅上的完整示例GitHub。

在Amazon SES 中验证电子邮件地址

以下示例演示了如何使用AWS适用于 SDK for Ruby 来验证 Amazon SES 电子邮件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
```

```
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses'  # v2: require 'aws-sdk'

# Replace recipient@example.com with a "To" address.
recipient = "recipient@example.com"

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to verify email address.
begin
  ses.verify_email_identity({
    email_address: recipient
  })

  puts 'Email sent to ' + recipient

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

请参阅上的完整示例GitHub。

向 Amazon SES 中的电子邮件地址发送消息

以下示例演示了如何使用AWS适用于 SDK for Ruby 向 Amazon SES 电子邮件地址发送消息。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses'  # v2: require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = 'sender@example.com'

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = 'recipient@example.com'

# Specify a configuration set. To use a configuration
# set, uncomment the next line and line 74.
#   configsetname = "ConfigSet"
```

```
# The subject line for the email.
subject = 'Amazon SES test (AWS SDK for Ruby)'

# The HTML body of the email.
htmlbody =
  '<h1>Amazon SES test (AWS SDK for Ruby)</h1>'\
  '<p>This email was sent with <a href="https://aws.amazon.com/ses/">'\
  'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">'\
  'AWS SDK for Ruby</a>.'

# The email body for recipients with non-HTML email clients.
textbody = 'This email was sent with Amazon SES using the AWS SDK for Ruby.'

# Specify the text encoding scheme.
encoding = 'UTF-8'

# Create a new SES client in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to send the email.
begin
  # Provide the contents of the email.
  ses.send_email(
    destination: {
      to_addresses: [
        recipient
      ]
    },
    message: {
      body: {
        html: {
          charset: encoding,
          data: htmlbody
        },
        text: {
          charset: encoding,
          data: textbody
        }
      },
      subject: {
        charset: encoding,
        data: subject
      }
    },
    source: sender,
    # Uncomment the following line to use a configuration set.
    # configuration_set_name: configsetname,
  )

  puts 'Email sent to ' + recipient


# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"
end
```

请参阅上的完整示例GitHub。

获取Amazon SES 统计数据

以下示例演示如何使用适用于 Ruby 的 SES AWS 开发工具包获取有关 Amazon SES 的统计数据。使用此类信息来避免在退回或拒绝电子邮件时损坏您的声誉。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-ses'  # v2: require 'aws-sdk'

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

begin
  # Get send statistics so we don't ruin our reputation
  resp = ses.get_send_statistics({})

  dps = resp.send_data_points

  puts "Got #{dps.count} data point(s):"
  puts

  dps.each do |dp|
    puts "Timestamp:  #{dp.timestamp}" #=> Time
    puts "Attempts:   #{dp.delivery_attempts}" #=> Integer
    puts "Bounces:    #{dp.bounces}" #=> Integer
    puts "Complaints: #{dp.complaints}" #=> Integer
    puts "Rejects:    #{dp.rejects}"  #-> Integer
    puts
  end

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts "Error: #{error}"
end
```

请参阅上的完整示例GitHub。

使用适用于 Ruby 的AWS软件开发工具包的Amazon SNS

Amazon Simple Notification Service (Amazon SS) 是一项 Web 服务，通过该服务，应用程序、终端用户和设备可以即时发送和接收云通知。您可以使用以下示例使用适用于 Ruby 的AWS软件开发工具包访问 Amazon SNS。有关 Amazon SNS 的更多信息，请参阅 Amazon SNS 文档。

主题

- 获取有关所有Amazon SNS 主题的信息 (p. 169)
- 创建 Amazon SNS 主题 (p. 169)
- 获取有关 Amazon SNS 主题中所有订阅的信息 (p. 169)
- 在 Amazon SNS 主题中创建订阅 (p. 170)
- 向所有 Amazon SNS 主题订阅者发送消息 (p. 170)
- 允许将资源发布到 Amazon SNS 主题 (p. 171)

获取有关所有Amazon SNS 主题的信息

以下示例列出了您在该us-west-2地区的 Amazon SNS 主题的 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

sns.topics.each do |topic|
  puts topic.arn
end
```

创建 Amazon SNS 主题

以下示例在 MyGroovyTopic 区域中创建主题 us-west-2 并显示生成的主题 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.create_topic(name: 'MyGroovyTopic')
puts topic.arn
```

获取有关 Amazon SNS 主题中所有订阅的信息

以下示例列出了该us-west-2区域中具有 ARN 的主题的 Amazon SNS 订阅arn:aws:sns:us-west-2:123456789:MyGroovyTopic的电子邮件地址。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
```

```
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

topic.subscriptions.each do |s|
  puts s.attributes['Endpoint']
end
```

在 Amazon SNS 主题中创建订阅

以下示例为 us-west-2 区域中具有电子邮件地址 MyGroovyUser@MyGroovy.com 的用户创建 ARN 为 arn:aws:sns:us-west-2:123456789:MyGroovyTopic 的主题的订阅并显示生成的 ARN。最初 ARN 值等待确认。在用户确认其电子邮件地址后，此值成为真正的 ARN。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

sub = topic.subscribe({
  protocol: 'email',
  endpoint: 'MyGroovyUser@MyGroovy.com'
})

puts sub.arn
```

向所有 Amazon SNS 主题订阅者发送消息

以下示例将消息 “Hello!” 发送给 适用于使用 ARN 的 Amazon SNS 主题的所有订阅者arn:aws:sns:us-west-2:123456789:MyGroovyTopic。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'
```

```
sns = Aws::SNS::Resource.new(region: 'us-west-2')

topic = sns.topic('arn:aws:sns:us-west-2:123456789:MyGroovyTopic')

topic.publish({
  message: 'Hello!'
})
```

允许将资源发布到 Amazon SNS 主题

以下示例允许 ARN 为 `my-resource-arn` 的资源发布到 `us-west-2` 区域中 ARN 为 `my-topic-arn` 的主题。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sns'  # v2: require 'aws-sdk'

policy  = '{
  "Version":"2008-10-17",
  "Id":"__default_policy_ID",
  "Statement":[{
    "Sid":"__default_statement_ID",
    "Effect":"Allow",
    "Principal":{
      "AWS":"*"
    },
    "Action":["SNS:Publish"],
    "Resource":"' + my-topic-arn + '",
    "Condition":{
      "ArnEquals":{
        "AWS:SourceArn":"' + my-resource-arn + '"}
     }
  }]
}'

sns = Aws::SNS::Resource.new(region: 'us-west-2')

# Get topic by ARN
topic = sns.topic(my-topic-arn)

# Add policy to topic
topic.set_attributes({
  attribute_name: "Policy",
  attribute_value: policy
})
```

使用适用于 S AWS DK for Ruby 的Amazon SQS 示例

Amazon Simple Queue Service (Amazon SQS) 是一种完全托管的消息队列服务，可让您轻松解耦和扩展微服务、分布式系统和无服务器应用程序。您可以使用以下示例使用适用于 Ruby 的软件开发工具包访问 Amazon AWS SQS。有关 Amazon SQS 的更多信息，请参阅 Amazon SQS 文档。

主题

- 获取有关 Amazon SQS 中所有队列的信息 (p. 172)
- 在Amazon SQS 中创建队列 (p. 173)
- 在Amazon SQS 中处理队列 (p. 174)
- 在Amazon SQS 中发送消息 (p. 175)
- 在Amazon SQS 中发送和接收消息 (p. 177)
- 在Amazon SQS 中接收消息 (p. 179)
- 在 Amazon SQS 中使用长轮询接收消息 (p. 180)
- 在Amazon SQS 中启用长轮询 (p. 181)
- 在 Amazon SQS 中使用QueuePoller类接收消息 (p. 183)
- 在Amazon SQS 中重定向死信 (p. 185)
- 在Amazon SQS 中删除队列 (p. 185)
- 在 Amazon SQS 中启用资源发布到队列 (p. 185)
- 在 Amazon SQS 中处理死信队列 (p. 186)
- 在 Amazon SQS 中指定消息可见性超时 (p. 188)

获取有关 Amazon SQS 中所有队列的信息

以下示例列出了您在该地区的 Amazon SQS 队列的 URL、ARN、可用消息和正在运行的`us-west-2`消息。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Lists the URLs of available queues in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @example
#   list_queue_urls(Aws::SQS::Client.new(region: 'us-east-1'))
def list_queue_urls(sqs_client)
  queues = sqs_client.list_queues

  queues.queue_urls.each do |url|
    puts url
  end
rescue StandardError => e
  puts "Error listing queue URLs: #{e.message}"
end

# Lists the attributes of a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @example
#   list_queue_attributes(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
#   )
def list_queue_attributes(sqs_client, queue_url)
  attributes = sqs_client.get_queue_attributes(
    queue_url: queue_url,
    attribute_names: [ "All" ]
  )

  attributes.attributes.each do |key, value|
```

```
    puts "#{key}: #{value}"
  end

rescue StandardError => e
  puts "Error getting queue attributes: #{e.message}"
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'

  sqs_client = Aws::SQS::Client.new(region: region)

  puts 'Listing available queue URLs...'
  list_queue_urls(sqs_client)

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  puts "\nGetting information about queue '#{queue_name}'..."
  list_queue_attributes(sqs_client, queue_url)
end

run_me if $PROGRAM_NAME == __FILE__
```

在Amazon SQS 中创建队列

以下示例创建了在us-west-2区域MyGroovyQueue中命名的 Amazon SQS 队列并显示其 URL。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'

# Creates a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_name [String] The name of the queue.
# @return [Boolean] true if the queue was created; otherwise, false.
# @example
#   exit 1 unless queue_created?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'my-queue'
#   )
def queue_created?(sqs_client, queue_name)
  sqs_client.create_queue(queue_name: queue_name)
  true
rescue StandardError => e
  puts "Error creating queue: #{e.message}"
  false
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Creating the queue named '#{queue_name}'..."
```

```
  if queue_created?(sqs_client, queue_name)
    puts 'Queue created.'
  else
    puts 'Queue not created.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在Amazon SQS 中处理队列

Amazon SQS 提供了高度可扩展的托管队列，用于存储应用程序或微服务之间传输的消息。要了解有关队列的更多信息，请参阅 Amazon SQS 队列的工作原理。

在此示例中，您使用带有 Amazon SQS 的 AWS SDK for Ruby 来：

1. 使用获取队列列表Aws::SQS::Client#list_queues。
2. 使用创建队列Aws::SQS::Client#create_queue。
3. 使用获取队列的 URL Aws::SQS::Client#get_queue_url。
4. 使用删除队列Aws::SQS::Client#delete_queue。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

示例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Get a list of your queues.
# 2. Create a queue.
# 3. Get the queue's URL.
# 4. Delete the queue.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Get a list of your queues.
sqs.list_queues.queue_urls.each do |queue_url|
  puts queue_url
end

# Create a queue.
```

```
queue_name = "my-queue"

begin
  sqs.create_queue({
    queue_name: queue_name,
    attributes: {
      "DelaySeconds" => "60", # Delay message delivery for 1 minute (60 seconds).
      "MessageRetentionPeriod" => "86400" # Delete message after 1 day (24 hours * 60
 minutes * 60 seconds).
    }
  })
rescue Aws::SQS::Errors::QueueDeletedRecently
  puts "A queue with the name '#{queue_name}' was recently deleted. Wait at least 60
 seconds and try again."
  exit(false)
end

# Get the queue's URL.
queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url
puts queue_url

# Delete the queue.
sqs.delete_queue(queue_url: queue_url)
```

在Amazon SQS 中发送消息

以下示例通过Amazon SQS 队列发送消息 “Hello world”，网址URL位于该us-west-2区域。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Sends a message to a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param message_body [String] The contents of the message to be sent.
# @return [Boolean] true if the message was sent; otherwise, false.
# @example
#   exit 1 unless message_sent?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     'This is my message.'
#   )
def message_sent?(sqs_client, queue_url, message_body)
  sqs_client.send_message(
    queue_url: queue_url,
    message_body: message_body
  )
  true
rescue StandardError => e
  puts "Error sending message: #{e.message}"
  false
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  message_body = 'This is my message.'

  sts_client = Aws::STS::Client.new(region: region)
```

```
  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending a message to the queue named '#{queue_name}'..."

  if message_sent?(sqs_client, queue_url, message_body)
    puts 'Message sent.'
  else
    puts 'Message not sent.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

以下示例发送消息 “Hello world” 和 “天气怎么样？” 使用该us-west-2地区的 URL URL 通过Amazon SQS 队列。

> **Note**
>
> 如果您的队列是 FIFO 队列，则除了 message_group_id 和 id 参数外，还必须包括 message_body 参数。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Sends multiple messages as a batch to a queue in
# Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param entries [Hash] The contents of the messages to be sent,
#   in the correct format.
# @return [Boolean] true if the messages were sent; otherwise, false.
# @example
#   exit 1 unless messages_sent?(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     [
#       {
#         id: 'Message1',
#         message_body: 'This is the first message.'
#       },
#       {
#         id: 'Message2',
#         message_body: 'This is the second message.'
#       }
#     ]
#   )
def messages_sent?(sqs_client, queue_url, entries)
  sqs_client.send_message_batch(
    queue_url: queue_url,
    entries: entries
  )
  true
rescue StandardError => e
  puts "Error sending messages: #{e.message}"
  false
```

```
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  entries = [
    {
      id: 'Message1',
      message_body: 'This is the first message.'
    },
    {
      id: 'Message2',
      message_body: 'This is the second message.'
    }
  ]

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending messages to the queue named '#{queue_name}'..."

  if messages_sent?(sqs_client, queue_url, entries)
    puts 'Messages sent.'
  else
    puts 'Messages not sent.'
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

在Amazon SQS 中发送和接收消息

在 Amazon SQS 中创建队列后，您可以向其发送消息，然后使用该队列。要了解更多信息，请参阅教程：将消息发送到 Amazon SQS 队列和教程：接收和删除来自 Amazon SQS 队列的消息。

在此示例中，您使用带有 Amazon SQS 的 AWS SDK for Ruby 来：

1. 通过使用 Aws::SQS::Client#send_message 来向队列发送消息。

   Note

   如果您的队列是 FIFO 队列，则除了 `message_group_id` 和 `id` 参数外，还必须包括 `message_body` 参数。

1. 通过使用 Aws::SQS::Client#receive_message 来接收队列中的消息。
2. 显示有关消息的信息。
3. 通过使用 Aws::SQS::Client#delete_message 来删除队列中的消息。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)

- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要创建队列 my-queue，您可以在 Amazon SQS 控制台中创建队列。

示例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Send a message to a queue.
# 2. Receive the message in the queue.
# 3. Display information about the message.
# 4. Delete the message from the queue.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Send a message to a queue.
queue_name = "my-queue"

begin
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  # Create a message with three custom attributes: Title, Author, and WeeksOn.
  send_message_result = sqs.send_message({
    queue_url: queue_url,
    message_body: "Information about current NY Times fiction bestseller for week of
 2016-12-11.",
    message_attributes: {
      "Title" => {
        string_value: "The Whistler",
        data_type: "String"
      },
      "Author" => {
        string_value: "John Grisham",
        data_type: "String"
      },
      "WeeksOn" => {
        string_value: "6",
        data_type: "Number"
      }
    }
  })
rescue Aws::SQS::Errors::NonExistentQueue
  puts "A queue named '#{queue_name}' does not exist."
  exit(false)
end

puts send_message_result.message_id

# Receive the message in the queue.
receive_message_result = sqs.receive_message({
  queue_url: queue_url,
```

```
  message_attribute_names: ["All"], # Receive all custom attributes.
  max_number_of_messages: 1, # Receive at most one message.
  wait_time_seconds: 0 # Do not wait to check for the message.
})

# Display information about the message.
# Display the message's body and each custom attribute value.
receive_message_result.messages.each do |message|
  puts message.body
  puts "Title: #{message.message_attributes["Title"]["string_value"]}"
  puts "Author: #{message.message_attributes["Author"]["string_value"]}"
  puts "WeeksOn: #{message.message_attributes["WeeksOn"]["string_value"]}"

  # Delete the message from the queue.
  sqs.delete_message({
    queue_url: queue_url,
    receipt_handle: message.receipt_handle
  })
end
```

在Amazon SQS 中接收消息

以下示例显示了 Amazon SQS 队列中最多 10 条消息的正文，URL URL 位于该us-west-2区域。

> **Note**
>
> receive_message 不保证获取所有消息 (请参阅分布式队列的属性)，并且默认情况下不会删除消息。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

require 'aws-sdk-sqs'
require 'aws-sdk-sts'

# Receives messages in a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param max_number_of_messages [Integer] The maximum number of messages
#   to receive. This number must be 10 or less. The default is 10.
# @example
#   receive_messages(
#     Aws::SQS::Client.new(region: 'us-east-1'),
#     'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue',
#     10
#   )
def receive_messages(sqs_client, queue_url, max_number_of_messages = 10)

  if max_number_of_messages > 10
    puts 'Maximum number of messages to receive must be 10 or less. ' \
      'Stopping program.'
    return
  end

  response = sqs_client.receive_message(
    queue_url: queue_url,
    max_number_of_messages: max_number_of_messages
  )

  if response.messages.count.zero?
    puts 'No messages to receive, or all messages have already ' \
      'been previously received.'
    return
```

```
  end

  response.messages.each do |message|
    puts '-' * 20
    puts "Message body: #{message.body}"
    puts "Message ID:   #{message.message_id}"
  end

rescue StandardError => e
  puts "Error receiving messages: #{e.message}"
end

# Full example call:
def run_me
  region = 'us-east-1'
  queue_name = 'my-queue'
  max_number_of_messages = 10

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-east-1.amazonaws.com/111111111111/my-queue'
  queue_url = 'https://sqs.' + region + '.amazonaws.com/' +
    sts_client.get_caller_identity.account + '/' + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Receiving messages from queue '#{queue_name}'..."

  receive_messages(sqs_client, queue_url, max_number_of_messages)
end

run_me if $PROGRAM_NAME == __FILE__
```

在 Amazon SQS 中使用长轮询接收消息

以下示例最多等待 10 秒钟，然后在 Amazon SQS 队列中显示最多 10 条消息的正文，网址URL位于该us-west-2区域。

如果您未指定等待时间，则默认值为 0（Amazon SQS 不等待）。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

resp = sqs.receive_message(queue_url: URL, max_number_of_messages: 10, wait_time_seconds:
 10)

resp.messages.each do |m|
  puts m.body
end
```

在Amazon SQS 中启用长轮询

长轮询通过减少空响应的数量和消除虚假的空响应来帮助您降低使用 Amazon SQS 的成本。有关长轮询的更多信息，请参阅 Amazon SQS 长轮询。

在此示例中，您使用带有 Amazon SQS 的 AWS SDK for Ruby 来：

1. 通过使用 Aws::SQS::Client#create_queue 来创建队列并将其设置为长轮询。
2. 通过使用 Aws::SQS::Client#set_queue_attributes 来为现有队列设置长轮询。
3. 通过使用 Aws::SQS::Client#receive_message 来设置接收队列的消息时的长轮询。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要创建现有队列和接收队列，您可以在 Amazon SQS 控制台中执行此操作。

示例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Create a queue and set it for long polling.
# 2. Set long polling for an existing queue.
# 3. Set long polling when receiving messages for a queue.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Create a queue and set it for long polling.
new_queue_name = "new-queue"

create_queue_result = sqs.create_queue({
  queue_name: new_queue_name,
  attributes: {
    "ReceiveMessageWaitTimeSeconds" => "20" # Wait 20 seconds to receive messages.
  },
})

puts create_queue_result.queue_url

# Set long polling for an existing queue.
begin
  existing_queue_name = "existing-queue"
  existing_queue_url = sqs.get_queue_url(queue_name: existing_queue_name).queue_url
```

```
  sqs.set_queue_attributes({
    queue_url: existing_queue_url,
    attributes: {
      "ReceiveMessageWaitTimeSeconds" => "20" # Wait 20 seconds to receive messages.
    },
  })
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot set long polling for a queue named '#{existing_queue_name}', as it does not
 exist."
end

# Set long polling when receiving messages for a queue.

# 1. Using receive_message.
begin
  receive_queue_name = "receive-queue"
  receive_queue_url = sqs.get_queue_url(queue_name: receive_queue_name).queue_url

  puts "Begin receipt of any messages using receive_message..."
  receive_message_result = sqs.receive_message({
    queue_url: receive_queue_url,
    attribute_names: ["All"], # Receive all available built-in message attributes.
    message_attribute_names: ["All"], # Receive any custom message attributes.
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Received #{receive_message_result.messages.count} message(s)."
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages using receive_message for a queue named
 '#{receive_queue_name}', as it does not exist."
end

# 2. Using Aws::SQS::QueuePoller.
begin
  puts "Begin receipt of any messages using Aws::SQS::QueuePoller..."
  puts "(Will keep polling until no more messages available for at least 60 seconds.)"
  poller = Aws::SQS::QueuePoller.new(receive_queue_url)

  poller_stats = poller.poll({
    max_number_of_messages: 10,
    idle_timeout: 60 # Stop polling after 60 seconds of no more messages available (polls
 indefinitely by default).
  }) do |messages|
    messages.each do |message|
      puts "Message body: #{message.body}"
    end
  end
  # Note: If poller.poll is successful, all received messages are automatically deleted
 from the queue.

  puts "Poller stats:"
  puts "  Polling started at: #{poller_stats.polling_started_at}"
  puts "  Polling stopped at: #{poller_stats.polling_stopped_at}"
  puts "  Last message received at: #{poller_stats.last_message_received_at}"
  puts "  Number of polling requests: #{poller_stats.request_count}"
  puts "  Number of received messages: #{poller_stats.received_message_count}"
rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages using Aws::SQS::QueuePoller for a queue named
 '#{receive_queue_name}', as it does not exist."
end
```

在 Amazon SQS 中使用QueuePoller类接收消息

以下示例使用QueuePoller实用程序类显示 Amazon SQS 队列中所有消息的正文以及该us-west-2区域URL中的 URL，并删除该消息。不活动时间达到大约 15 秒后，脚本超时。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(idle_timeout: 15) do |msg|
  puts msg.body
end
```

以下示例使用 URL 循环浏览 Amazon SQS 队列URL，最多等待几秒钟。

您可以通过执行 Amazon SQS 中获取有关所有队列的信息中的 Amazon SQS 示例来获得 (p. 172)正确的URL。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(wait_time_seconds: duration, idle_timeout: duration + 1) do |msg|
  puts msg.body
end
```

以下示例使用 URL URL 循环遍历 Amazon SQS 队列，并为您提供处理消息的可见性超时秒数，由方法do_something表示。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
```

```
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

# Process the message
def do_something(msg)
  puts msg.body
end

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(visibility_timeout: timeout, idle_timeout: timeout + 1) do |msg|
  do_something(msg)
end
```

以下示例使用 URL URL 循环遍历 Amazon SQS 队列，并更改任何需要该方法do_something2额外处理的消息的可见性超时秒数。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

# Process the message
def do_something(_)
  true
end

# Do additional processing
def do_something2(msg)
  puts msg.body
end

Aws.config.update({region: 'us-west-2'})

poller = Aws::SQS::QueuePoller.new(URL)

poller.poll(idle_timeout: timeout + 1) do |msg|
  if do_something(msg)
    # need more time for processing
    poller.change_message_visibility_timeout(msg, timeout)

    do_something2(msg)
  end
end
```

在Amazon SQS 中重定向死信

以下示例将 URL 为 URL 的队列中的任何死信重定向到 ARN 为 ARN 的队列。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

sqs.set_queue_attributes({
  queue_url: URL,
  attributes:
    {
      'RedrivePolicy' => "{\"maxReceiveCount\":\"5\", \"deadLetterTargetArn\":\"#{ARN}\"}"
    }
})
```

在Amazon SQS 中删除队列

以下示例删除 URL URL 位于该us-west-2区域的 Amazon SQS 队列。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

sqs.delete_queue(queue_url: URL)
```

在 Amazon SQS 中启用资源发布到队列

以下示例允许 ARN 为 my-resource-arn 的资源发布到 my-queue-arn 区域中 ARN 为 my-queue-url 且 URL 为 us-west-2 的队列。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
```

```
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-west-2')

policy  = '{
  "Version":"2008-10-17",
  "Id":' + my-queue-arn + '/SQSDefaultPolicy",
  "Statement":[{
    "Sid":"__default_statement_ID",
    "Effect":"Allow",
    "Principal":{
      "AWS":"*"
    },
    "Action":["SQS:SendMessage"],
    "Resource":"' + my-queue-arn + '",
    "Condition":{
      "ArnEquals":{
        "AWS:SourceArn":"' + my-resource-arn + '"}
     }
  }]
}'

sqs.set_queue_attributes({
  queue_url: my-queue-url,
  attributes: {
    Policy: policy
  }
})
```

在 Amazon SQS 中处理死信队列

Amazon SQS 为死信队列提供支持。死信队列是其他（源）队列可将其作为无法成功处理的消息的目标的队列。您可以搁置和隔离死信队列中的这些消息以确定其处理失败的原因。有关死信队列的更多信息，请参阅使用 Amazon SQS 死信队列。

在此示例中，您使用带有 Amazon SQS 的 AWS SDK for Ruby 来：

1. 通过使用 Aws::SQS::Client#create_queue 来创建表示死信队列的队列。
2. 通过使用 Aws::SQS::Client#set_queue_attributes 来将死信队列与现有队列相关联。
3. 通过使用 Aws::SQS::Client#send_message 来向现有队列发送消息。
4. 使用 Aws:: SQS:: 轮询队列。QueuePoller
5. 通过使用 Aws::SQS::Client#receive_message 来接收死信队列中的消息。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要使用 AWS Management Console创建现有队列 my-queue。

Note

为简单起见，此示例代码没有演示 Aws::SQS::Client#add_permission。在现实世界中，您应始终限制对SendMessage、ReceiveMessageDeleteMessage、和等操作的访问权限DeleteQueue。不这样做可能会导致信息泄露、拒绝服务或将消息注入您的队列中。

示例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to:
# 1. Create a queue representing a dead letter queue.
# 2. Associate the dead letter queue with an existing queue.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

# Uncomment for Windows.
# Aws.use_bundled_cert!

sqs = Aws::SQS::Client.new(region: 'us-east-1')

# Create a queue representing a dead letter queue.
dead_letter_queue_name = "dead-letter-queue"

sqs.create_queue({
  queue_name: dead_letter_queue_name
})

# Get the dead letter queue's URL and ARN, so that you can associate it with an existing
 queue.
dead_letter_queue_url = sqs.get_queue_url(queue_name: dead_letter_queue_name).queue_url

dead_letter_queue_arn = sqs.get_queue_attributes({
  queue_url: dead_letter_queue_url,
  attribute_names: ["QueueArn"]
}).attributes["QueueArn"]

# Associate the dead letter queue with an existing queue.
begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  # Use a redrive policy to specify the dead letter queue and its behavior.
  redrive_policy = {
    "maxReceiveCount" => "5", # After the queue receives the same message 5 times, send
 that message to the dead letter queue.
    "deadLetterTargetArn" => dead_letter_queue_arn
  }.to_json

  sqs.set_queue_attributes({
    queue_url: queue_url,
    attributes: {
      "RedrivePolicy" => redrive_policy
    }
```

```
  })

rescue Aws::SQS::Errors::NonExistentQueue
  puts "A queue named '#{queue_name}' does not exist."
  exit(false)
end

# Send a message to the queue.
puts "Sending a message..."

sqs.send_message({
  queue_url: queue_url,
  message_body: "I hope I get moved to the dead letter queue."
})

30.downto(0) do |i|
  print "\rWaiting #{i} second(s) for sent message to be receivable..."
  sleep(1)
end

puts "\n"

poller = Aws::SQS::QueuePoller.new(queue_url)
# Receive 5 messages max and stop polling after 20 seconds of no received messages.
poller.poll(max_number_of_messages:5, idle_timeout: 20) do |messages|
  messages.each do |msg|
    puts "Received message ID: #{msg.message_id}"
  end
end

# Check to see if Amazon SQS moved the message to the dead letter queue.
receive_message_result = sqs.receive_message({
  queue_url: dead_letter_queue_url,
  max_number_of_messages: 1
})

if receive_message_result.messages.count > 0
  puts "\n#{receive_message_result.messages[0].body}"
else
  puts "\nNo messages received."
end
```

在 Amazon SQS 中指定消息可见性超时

在 Amazon SQS 中，收到消息后，该消息立即保留在队列中。为了防止其他消费者再次处理消息，Amazon SQS 设置了可见性超时。这段时间内 Amazon SQS 阻止其他消耗组件接收和处理消息。要了解更多信息，请参阅可见性超时。

在此示例中，您使用带有 Amazon SQS 的 AWS SDK for Ruby 来：

1. 通过使用 Aws::SQS::Client#get_queue_url 来获取现有队列的 URL。
2. 通过使用 Aws::SQS::Client#receive_message 来接收最多 10 个消息。
3. 通过使用 Aws::SQS::Client#change_message_visibility 来指定在收到消息后消息不可见的时间间隔。

先决条件

在运行示例代码之前，您需要安装和配置 AWS SDK for Ruby，如中所述：

- 安装AWS适用于 SDK for Ruby (p. 3)
- 为 Ruby 配置AWS软件开发工具包 (p. 9)

您还需要创建队列 my-queue，您可以在 Amazon SQS 控制台中创建队列。

示例

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

# Demonstrates how to specify the time interval during which messages to a queue are not
 visible after being received.

require 'aws-sdk-sqs'  # v2: require 'aws-sdk'

sqs = Aws::SQS::Client.new(region: 'us-east-1')

begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  receive_message_result_before = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Before attempting to change message visibility timeout: received
 #{receive_message_result_before.messages.count} message(s)."

  receive_message_result_before.messages.each do |message|
    sqs.change_message_visibility({
      queue_url: queue_url,
      receipt_handle: message.receipt_handle,
      visibility_timeout: 30 # This message will not be visible for 30 seconds after first
 receipt.
    })
  end

  # Try to retrieve the original messages after setting their visibility timeout.
  receive_message_result_after = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10
  })

  puts "\nAfter attempting to change message visibility timeout: received
 #{receive_message_result_after.messages.count} message(s)."

rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages for a queue named '#{receive_queue_name}', as it does not
 exist."
end
```

亚马逊WorkDocs示例

您可以使用以下示例使用AWS适用于 SDK for Ruby 访问亚马逊WorkDocs（亚马逊WorkDocs）。有关亚马逊的更多信息WorkDocs，请参阅亚马逊WorkDocs文档。

您需要您的组织 ID 才能使用这些示例。使用以下步骤从AWS控制台获取您的组织 ID：

- 选择 AWS Directory Service
- 选择 `Directories`

组织 ID 与您的亚马逊WorkDocs网站`Directory ID`相对应。

示例

主题

- 列出用户 (p. 190)
- 列出用户文档 (p. 191)

列出用户

以下示例列出了组织中的所有用户的名称、电子邮件地址和根文件夹。选择 Copy 将代码保存在本地，或参阅本主题末尾的完整示例的链接。

1. 需要 S AWS DK for Ruby 模块并创建亚马逊WorkDocs客户端。

2. 使用您的组织 ID 调用 `describe_users`，并按升序获取所有用户名。

1. 显示有关用户的信息。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-workdocs'  # v2: require 'aws-sdk'

client = Aws::WorkDocs::Client.new(region: 'us-west-2')

# Set to the OrganizationId of your WorkDocs site
orgId = 'd-123456789c'

resp = client.describe_users({
  organization_id: orgId,
  include: "ALL", # accepts ALL, ACTIVE_PENDING
  order: "ASCENDING", # accepts ASCENDING, DESCENDING
  sort: "USER_NAME", # accepts USER_NAME, FULL_NAME, STORAGE_LIMIT, USER_STATUS,
 STORAGE_USED
})

resp.users.each do |user|
```

```
  puts "First name:  #{user.given_name}"
  puts "Last name:   #{user.surname}"
  puts "Email:       #{user.email_address}"
  puts "Root folder: #{user.root_folder_id}"
  puts
end
```

请参阅上的完整示例GitHub。

列出用户文档

以下示例列出用户的文档。选择 Copy 将代码保存在本地，或参阅本主题末尾的完整示例的链接。

1. 需要适用于 Ruby 模块的AWS软件开发工具包。
2. 创建一个帮助程序方法来获取用户的根文件夹。
3. 创建亚马逊WorkDocs客户端。
4. 获取该用户的根文件夹。
5. 调用 `describe_folder_contents` 以按升序获取文件夹的内容。
6. 显示用户的根文件夹中的每个文档的名称、大小（以字节为单位）、上次修改日期、文档 ID 和版本 ID。

```
# Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# This file is licensed under the Apache License, Version 2.0 (the "License").
# You may not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, either express or implied. See the License for the specific
# language governing permissions and limitations under the License.

require 'aws-sdk-workdocs'  # v2: require 'aws-sdk'

def get_user_folder(client, orgId, user_email)
  root_folder = ''

  resp = client.describe_users({
    organization_id: orgId,
  })

  # resp.users should have only one entry
  resp.users.each do |user|
    if user.email_address == user_email
      root_folder = user.root_folder_id
    end
  end

  return root_folder
end

client = Aws::WorkDocs::Client.new(region: 'us-west-2')

# Set to the email address of a user
user_email = 'someone@somewhere'

# Set to the OrganizationId of your WorkDocs site.
orgId = 'd-123456789c'
```

```
user_folder = get_user_folder(client, orgId, user_email)

if user_folder == ''
  puts 'Could not get root folder for user with email address ' + user_email
  exit(1)
end

resp = client.describe_folder_contents({
  folder_id: user_folder, # required
  sort: "NAME", # accepts DATE, NAME
  order: "ASCENDING", # accepts ASCENDING, DESCENDING
})

resp.documents.each do |doc|
  md = doc.latest_version_metadata

  puts "Name:          #{md.name}"
  puts "Size (bytes):  #{md.size}"
  puts "Last modified: #{doc.modified_timestamp}"
  puts "Doc ID:        #{doc.id}"
  puts "Version ID:    #{md.id}"
  puts
end
```

请参阅上的完整示例GitHub。

适用于 Ruby 的软件开发工具包代码

本主题中的代码示例向您展示了如何使用 wit AWS SDK for Ruby h AWS。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

跨服务示例是指跨多个 AWS 服务 工作的示例应用程序。

示例

- 使用适用于 Ruby 的 SDK 的操作和场景 (p. 193)

使用适用于 Ruby 的 SDK 的操作和场景

以下代码示例说明如何使用AWS SDK for Ruby与执行操作和实现常见场景AWS 服务。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

服务

- 使用 Ruby 开发工具包的 DynamoDB 示例 (p. 193)
- EventBridge使用 Ruby 软件开发工具包的示例 (p. 210)
- AWS Glue使用 Ruby 软件开发工具包的示例 (p. 224)
- 使用 Ruby 开发工具包的 IAM 示例 (p. 242)
- 使用 Ruby 开发工具包的 Lambda 示例 (p. 261)
- 使用 Ruby 开发工具包的亚马逊 RDS 示例 (p. 271)
- 使用 Ruby 软件开发工具包的亚马逊 S3 示例 (p. 274)
- 使用 Ruby 开发工具包的亚马逊 SNS 示例 (p. 293)
- 使用适用于 Ruby 开发工具包的亚马逊 SQS 示例 (p. 297)
- AWS STS使用 Ruby 软件开发工具包的示例 (p. 305)

使用 Ruby 开发工具包的 DynamoDB 示例

以下代码示例向您展示了如何使用与 DynamoDB AWS SDK for Ruby 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)

- 场景 (p. 205)

操作

创建表

以下代码示例显示了如何创建 DynamoDB 表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Creates an Amazon DynamoDB table that can be used to store movie data.
  # The table uses the release year of the movie as the partition key and the
  # title as the sort key.
  #
  # @param table_name [String] The name of the table to create.
  # @return [Aws::DynamoDB::Table] The newly created table.
  def create_table(table_name)
    @table = @dynamo_resource.create_table(
      table_name: table_name,
      key_schema: [
        {attribute_name: "year", key_type: "HASH"},  # Partition key
        {attribute_name: "title", key_type: "RANGE"}  # Sort key
      ],
      attribute_definitions: [
        {attribute_name: "year", attribute_type: "N"},
        {attribute_name: "title", attribute_type: "S"}
      ],
      provisioned_throughput: {read_capacity_units: 10, write_capacity_units: 10})
    @dynamo_resource.client.wait_until(:table_exists, table_name: table_name)
    @table
  rescue Aws::DynamoDB::Errors::ServiceError => e
    @logger.error("Failed create table #{table_name}:\n#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 CreateTableAWS SDK for RubyAPI 参考文档。

删除表

以下代码示例显示了如何删除 DynamoDB 表。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Deletes the table.
  def delete_table
    @table.delete
    @table = nil
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't delete table. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DeleteTableAWS SDK for RubyAPI 参考文档。

从表中删除项目

以下代码示例显示了如何从 DynamoDB 表中删除项目。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Deletes a movie from the table.
  #
  # @param title [String] The title of the movie to delete.
  # @param year [Integer] The release year of the movie to delete.
  def delete_item(title, year)
    @table.delete_item(key: {"year" => year, "title" => title})
  rescue Aws::DynamoDB::Errors::ServiceError => e
```

```
    puts("Couldn't delete movie #{title}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DeleteItemAWS SDK for RubyAPI 参考文档。

从表中获取项目

以下代码示例显示了如何从 DynamoDB 表中获取项目。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Gets movie data from the table for a specific movie.
  #
  # @param title [String] The title of the movie.
  # @param year [Integer] The release year of the movie.
  # @return [Hash] The data about the requested movie.
  def get_item(title, year)
    @table.get_item(key: {"year" => year, "title" => title})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't get movie #{title} (#{year}) from table #{@table.name}:\n")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 GetItemAWS SDK for RubyAPI 参考文档。

有关表的信息

以下代码示例显示了如何获取有关 DynamoDB 表的信息。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table
```

```
  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Determines whether a table exists. As a side effect, stores the table in
  # a member variable.
  #
  # @param table_name [String] The name of the table to check.
  # @return [Boolean] True when the table exists; otherwise, False.
  def exists?(table_name)
    @dynamo_resource.client.describe_table(table_name: table_name)
    @logger.debug("Table #{table_name} exists")
  rescue Aws::DynamoDB::Errors::ResourceNotFoundException
    @logger.debug("Table #{table_name} doesn't exist")
    false
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't check for existence of #{table_name}:\n")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DescribeTableAWS SDK for RubyAPI 参考文档。

列出表

以下代码示例显示了如何列出 DynamoDB 表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

确定表是否存在。

```
# Encapsulates an Amazon DynamoDB table of movie data.
class Scaffold
  attr_reader :dynamo_resource
  attr_reader :table_name
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table_name = table_name
    @table = nil
    @logger = Logger.new($stdout)
    @logger.level = Logger::DEBUG
  end

  # Determines whether a table exists. As a side effect, stores the table in
  # a member variable.
  #
  # @param table_name [String] The name of the table to check.
  # @return [Boolean] True when the table exists; otherwise, False.
  def exists?(table_name)
```

```
    @dynamo_resource.client.describe_table(table_name: table_name)
    @logger.debug("Table #{table_name} exists")
  rescue Aws::DynamoDB::Errors::ResourceNotFoundException
    @logger.debug("Table #{table_name} doesn't exist")
    false
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't check for existence of #{table_name}:\n")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 ListTablesAWS SDK for RubyAPI 参考文档。

在表中放置项目

以下代码示例显示了如何将项目放入 DynamoDB 表中。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Adds a movie to the table.
  #
  # @param movie [Hash] The title, year, plot, and rating of the movie.
  def add_item(movie)
    @table.put_item(
      item: {
        "year" => movie[:year],
        "title" => movie[:title],
        "info" => {"plot" => movie[:plot], "rating" => movie[:rating]}})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't add movie #{title} to table #{@table.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 PutItemAWS SDK for RubyAPI 参考文档。

查询表

以下代码示例显示了如何查询 DynamoDB 表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Queries for movies that were released in the specified year.
  #
  # @param year [Integer] The year to query.
  # @return [Array] The list of movies that were released in the specified year.
  def query_items(year)
    response = @table.query(
      key_condition_expression: "#yr = :year",
      expression_attribute_names: {"#yr" => "year"},
      expression_attribute_values: {":year" => year})
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't query for movies released in #{year}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    response.items
  end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 Query。

运行 PartiQL 语句

以下代码示例显示了如何在 DynamoDB 表上运行 PartiQL 语句。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

使用 PartiQL 选择单个项目。

```
class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Gets a single record from a table using PartiQL.
  # Note: To perform more fine-grained selects,
  # use the Client.query instance method instead.
  #
  # @param title [String] The title of the movie to search.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def select_item_by_title(title)
    request = {
```

```
      statement: "SELECT * FROM \"#{@table.name}\" WHERE title=?",
      parameters: [title]
    }
    @dynamodb.client.execute_statement(request)
  end
```

使用 PartiQL 更新单个项目。

```
class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Updates a single record from a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @param rating [Float] The new rating to assign the title.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def update_rating_by_title(title, year, rating)
    request = {
      statement: "UPDATE \"#{@table.name}\" SET info.rating=? WHERE title=? and
 year=?",
      parameters: [{ "N": rating }, title, year]
    }
    @dynamodb.client.execute_statement(request)
  end
```

使用 PartiQL 添加单个项目。

```
class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Adds a single record to a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @param plot [String] The plot of the movie.
  # @param rating [Float] The new rating to assign the title.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def insert_item(title, year, plot, rating)
    request = {
      statement: "INSERT INTO \"#{@table.name}\" VALUE {'title': ?, 'year': ?,
 'info': ?}",
      parameters: [title, year, {'plot': plot, 'rating': rating}]
    }
    @dynamodb.client.execute_statement(request)
```

```
  end
```

使用 PartiQL 删除单个项目。

```
class DynamoDBPartiQLSingle

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Deletes a single record from a table using PartiQL.
  #
  # @param title [String] The title of the movie to update.
  # @param year [Integer] The year the movie was released.
  # @return [Aws::DynamoDB::Types::ExecuteStatementOutput]
  def delete_item_by_title(title, year)
    request = {
      statement: "DELETE FROM \"#{@table.name}\" WHERE title=? and year=?",
      parameters: [title, year]
    }
    @dynamodb.client.execute_statement(request)
  end
```

- 有关 API 的详细信息，请参阅 ExecuteStatementAWS SDK for RubyAPI 参考文档。

运行批量 PartiQL 语句

以下代码示例显示了如何在 DynamoDB 表上批量运行 PartiQL 语句。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

使用 PartiQL 读取一批项目。

```
class DynamoDBPartiQLBatch

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Selects a batch of items from a table using PartiQL
  #
  # @param batch_titles [Array] Collection of movie titles
  # @return [Aws::DynamoDB::Types::BatchExecuteStatementOutput]
  def batch_execute_select(batch_titles)
    request_items = batch_titles.map do |title, year|
```

```
      {
        statement: "SELECT * FROM \"#{@table.name}\" WHERE title=? and year=?",
        parameters: [title, year]
      }
    end
    @dynamodb.client.batch_execute_statement({statements: request_items})
  end
```

使用 PartiQL 删除一批项目。

```
class DynamoDBPartiQLBatch

  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamodb = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamodb.table(table_name)
  end

  # Deletes a batch of items from a table using PartiQL
  #
  # @param batch_titles [Array] Collection of movie titles
  # @return [Aws::DynamoDB::Types::BatchExecuteStatementOutput]
  def batch_execute_write(batch_titles)
    request_items = batch_titles.map do |title, year|
      {
        statement: "DELETE FROM \"#{@table.name}\" WHERE title=? and year=?",
        parameters: [title, year]
      }
    end
    @dynamodb.client.batch_execute_statement({statements: request_items})
  end
```

- 有关 API 的详细信息，请参阅 BatchExecuteStatementAWS SDK for RubyAPI 参考文档。

扫描表

以下代码示例显示了如何扫描 DynamoDB 表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Scans for movies that were released in a range of years.
  # Uses a projection expression to return a subset of data for each movie.
```

```
  #
  # @param year_range [Hash] The range of years to retrieve.
  # @return [Array] The list of movies released in the specified years.
  def scan_items(year_range)
    movies = []
    scan_hash = {
      filter_expression: "#yr between :start_yr and :end_yr",
      projection_expression: "#yr, title, info.rating",
      expression_attribute_names: {"#yr" => "year"},
      expression_attribute_values: {
        ":start_yr" => year_range[:start], ":end_yr" => year_range[:end]}
    }
    done = false
    start_key = nil
    until done
      scan_hash[:exclusive_start_key] = start_key unless start_key.nil?
      response = @table.scan(scan_hash)
      movies.concat(response.items) unless response.items.nil?
      start_key = response.last_evaluated_key
      done = start_key.nil?
    end
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't scan for movies. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    movies
  end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 Scan。

更新表中的项目

以下代码示例显示了如何更新 DynamoDB 表中的项目。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Updates rating and plot data for a movie in the table.
  #
  # @param movie [Hash] The title, year, plot, rating of the movie.
  def update_item(movie)

    response = @table.update_item(
      key: {"year" => movie[:year], "title" => movie[:title]},
      update_expression: "set info.rating=:r",
      expression_attribute_values: { ":r" => movie[:rating] },
      return_values: "UPDATED_NEW")
```

```
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts("Couldn't update movie #{movie[:title]} (#{movie[:year]}) in table
 #{@table.name}\n")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    response.attributes
  end
```

- 有关 API 的详细信息，请参阅 UpdateItemAWS SDK for RubyAPI 参考文档。

写入一批项目

以下代码示例显示了如何编写一批 DynamoDB 项目。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class DynamoDBBasics
  attr_reader :dynamo_resource
  attr_reader :table

  def initialize(table_name)
    client = Aws::DynamoDB::Client.new(region: "us-east-1")
    @dynamo_resource = Aws::DynamoDB::Resource.new(client: client)
    @table = @dynamo_resource.table(table_name)
  end

  # Fills an Amazon DynamoDB table with the specified data. Items are sent in
  # batches of 25 until all items are written.
  #
  # @param movies [Enumerable] The data to put in the table. Each item must contain at
 least
  #                            the keys required by the schema that was specified when
 the
  #                            table was created.
  def write_batch(movies)
    index = 0
    slice_size = 25
    while index < movies.length
      movie_items = []
      movies[index, slice_size].each do |movie|
        movie_items.append({put_request: { item: movie }})
      end
      @dynamo_resource.client.batch_write_item({request_items: { @table.name =>
 movie_items }})
      index += slice_size
    end
  rescue Aws::DynamoDB::Errors::ServiceError => e
    puts(
      "Couldn't load data into table #{@table.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 BatchWriteItemAWS SDK for RubyAPI 参考文档。

场景

开始使用表、项目和查询

以下代码示例显示了如何：

- 创建一个可以保存电影数据的表。
- 放置、获取和更新表中的单部电影。
- 从示例 JSON 文件将电影数据写入表中。
- 查询在给定年份发行的电影。
- 扫描几年内发行的电影。
- 从表中删除电影，然后删除表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

创建封装 DynamoDB 表的类。

```
# Creates an Amazon DynamoDB table that can be used to store movie data.
# The table uses the release year of the movie as the partition key and the
# title as the sort key.
#
# @param table_name [String] The name of the table to create.
# @return [Aws::DynamoDB::Table] The newly created table.
def create_table(table_name)
  @table = @dynamo_resource.create_table(
    table_name: table_name,
    key_schema: [
      {attribute_name: "year", key_type: "HASH"},  # Partition key
      {attribute_name: "title", key_type: "RANGE"}  # Sort key
    ],
    attribute_definitions: [
      {attribute_name: "year", attribute_type: "N"},
      {attribute_name: "title", attribute_type: "S"}
    ],
    provisioned_throughput: {read_capacity_units: 10, write_capacity_units: 10})
  @dynamo_resource.client.wait_until(:table_exists, table_name: table_name)
  @table
rescue Aws::DynamoDB::Errors::ServiceError => e
  @logger.error("Failed create table #{table_name}:\n#{e.code}: #{e.message}")
  raise
end
```

创建帮助函数以下载并提取示例 JSON 文件。

```
# Gets sample movie data, either from a local file or by first downloading it from
# the Amazon DynamoDB Developer Guide.
#
# @param movie_file_name [String] The local file name where the movie data is stored
in JSON format.
# @return [Hash] The movie data as a Hash.
def fetch_movie_data(movie_file_name)
  if !File.file?(movie_file_name)
    @logger.debug("Downloading #{movie_file_name}...")
    movie_content = URI.open(
```

```
        "https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/samples/
moviedata.zip"
      )
      movie_json = ""
      Zip::File.open_buffer(movie_content) do |zip|
        zip.each do |entry|
          movie_json = entry.get_input_stream.read
        end
      end
    else
      movie_json = File.read(movie_file_name)
    end
    movie_data = JSON.parse(movie_json)
    # The sample file lists over 4000 movies. This returns only the first 250.
    movie_data.slice(0, 250)
  rescue StandardError => e
    puts("Failure downloading movie data:\n#{e}")
    raise
  end
```

运行交互式场景以创建表并对其执行操作。

```
  table_name = "doc-example-table-movies-#{rand(10**4)}"
  scaffold = Scaffold.new(table_name)
  dynamodb_wrapper = DynamoDBBasics.new(table_name)

  new_step(1, "Create a new DynamoDB table if none already exists.")
  unless scaffold.exists?(table_name)
    puts("\nNo such table: #{table_name}. Creating it...")
    scaffold.create_table(table_name)
    print "Done!\n".green
  end

  new_step(2, "Add a new record to the DynamoDB table.")
  my_movie = {}
  my_movie[:title] = CLI::UI::Prompt.ask("Enter the title of a movie to add to the
table. E.g. The Matrix")
  my_movie[:year] = CLI::UI::Prompt.ask("What year was it released? E.g. 1989").to_i
  my_movie[:rating] = CLI::UI::Prompt.ask("On a scale of 1 - 10, how do you rate it?
E.g. 7").to_i
  my_movie[:plot] = CLI::UI::Prompt.ask("Enter a brief summary of the plot. E.g. A man
awakens to a new reality.")
  dynamodb_wrapper.add_item(my_movie)
  puts("\nNew record added:")
  puts JSON.pretty_generate(my_movie).green
  print "Done!\n".green

  new_step(3, "Update a record in the DynamoDB table.")
  my_movie[:rating] = CLI::UI::Prompt.ask("Let's update the movie you added with a new
rating, e.g. 3:").to_i
  response = dynamodb_wrapper.update_item(my_movie)
  puts("Updated '#{my_movie[:title]}' with new attributes:")
  puts JSON.pretty_generate(response).green
  print "Done!\n".green

  new_step(4, "Get a record from the DynamoDB table.")
  puts("Searching for #{my_movie[:title]} (#{my_movie[:year]})...")
  response = dynamodb_wrapper.get_item(my_movie[:title], my_movie[:year])
  puts JSON.pretty_generate(response).green
  print "Done!\n".green

  new_step(5, "Write a batch of items into the DynamoDB table.")
  download_file = "moviedata.json"
  puts("Downloading movie database to #{download_file}...")
```

```
  movie_data = fetch_movie_data(download_file)
  puts("Writing movie data from #{download_file} into your table...")
  scaffold.write_batch(movie_data)
  puts("Records added: #{movie_data.length}.")
  print "Done!\n".green

  new_step(5, "Query for a batch of items by key.")
  loop do
    release_year = CLI::UI::Prompt.ask("Enter a year between 1972 and 2018, e.g.
 1999:").to_i
    results = dynamodb_wrapper.query_items(release_year)
    if results.any?
      puts("There were #{results.length} movies released in #{release_year}:")
      results.each do |movie|
        puts("\t #{movie["title"]}").green
      end
      break
    else
      continue = CLI::UI::Prompt.ask("Found no movies released in #{release_year}! Try
 another year? (y/n)")
      break if !continue.eql?("y")
    end
  end
  print "\nDone!\n".green

  new_step(6, "Scan for a batch of items using a filter expression.")
  years = {}
  years[:start] = CLI::UI::Prompt.ask("Enter a starting year between 1972 and 2018:")
  years[:end] = CLI::UI::Prompt.ask("Enter an ending year between 1972 and 2018:")
  releases = dynamodb_wrapper.scan_items(years)
  if !releases.empty?
    puts("Found #{releases.length} movies.")
    count = Question.ask(
      "How many do you want to see? ", method(:is_int), in_range(1, releases.length))
    puts("Here are your #{count} movies:")
    releases.take(count).each do |release|
      puts("\t#{release["title"]}")
    end
  else
    puts("I don't know about any movies released between #{years[:start]} "\
         "and #{years[:end]}.")
  end
  print "\nDone!\n".green

  new_step(7, "Delete an item from the DynamoDB table.")
  answer = CLI::UI::Prompt.ask("Do you want to remove '#{my_movie[:title]}'? (y/n) ")
  if answer.eql?("y")
    dynamodb_wrapper.delete_item(my_movie[:title], my_movie[:year])
    puts("Removed '#{my_movie[:title]}' from the table.")
    print "\nDone!\n".green
  end

  new_step(8, "Delete the DynamoDB table.")
  answer = CLI::UI::Prompt.ask("Delete the table? (y/n)")
  if answer.eql?("y")
    dynamodb_wrapper.delete_table
    puts("Deleted #{table_name}.")
  else
    puts("Don't forget to delete the table when you're done!")
  end
  print "\nThanks for watching!\n".green
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo.")
rescue Errno::ENOENT
  true
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - 查询
  - Scan
  - UpdateItem

使用批量 PartIQL 语句查询表

以下代码示例显示了如何：

- 通过运行多个 SELECT 语句获取一批项目。
- 通过运行多个 INSERT 语句添加一批项目。
- 通过运行多个 UPDATE 语句更新一批项目。
- 通过运行多个 DELETE 语句删除一批项目。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

运行创建表并运行批量 PartIQL 查询的场景。

```
  table_name = "doc-example-table-movies-partiql-#{rand(10**4)}"
  scaffold = Scaffold.new(table_name)
  sdk = DynamoDBPartiQLBatch.new(table_name)

  new_step(1, "Create a new DynamoDB table if none already exists.")
  unless scaffold.exists?(table_name)
    puts("\nNo such table: #{table_name}. Creating it...")
    scaffold.create_table(table_name)
    print "Done!\n".green
  end

  new_step(2, "Populate DynamoDB table with movie data.")
  download_file = "moviedata.json"
  puts("Downloading movie database to #{download_file}...")
  movie_data = scaffold.fetch_movie_data(download_file)
  puts("Writing movie data from #{download_file} into your table...")
  scaffold.write_batch(movie_data)
  puts("Records added: #{movie_data.length}.")
  print "Done!\n".green

  new_step(3, "Select a batch of items from the movies table.")
  puts "Let's select some popular movies for side-by-side comparison."
  response = sdk.batch_execute_select([["Mean Girls", 2004], ["Goodfellas", 1977],
["The Prancing of the Lambs", 2005]])
  puts("Items selected: #{response['responses'].length}\n")
  print "\nDone!\n".green
```

```
  new_step(4, "Delete a batch of items from the movies table.")
  sdk.batch_execute_write([["Mean Girls", 2004], ["Goodfellas", 1977], ["The Prancing
 of the Lambs", 2005]])
  print "\nDone!\n".green

  new_step(5, "Delete the table.")
  if scaffold.exists?(table_name)
    scaffold.delete_table
  end
end
```

- 有关 API 的详细信息，请参阅 BatchExecuteStatementAWS SDK for RubyAPI 参考文档。

使用 PartiQL 查询表

以下代码示例显示了如何：

- 通过运行一个 SELECT 语句获取一个项目。
- 通过运行一个 INSERT 语句添加一个项目。
- 通过运行一个 UPDATE 语句更新一个项目。
- 通过运行一个 DELETE 语句删除一个项目。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

运行创建表并运行 PartIQL 查询的场景。

```
  table_name = "doc-example-table-movies-partiql-#{rand(10**8)}"
  scaffold = Scaffold.new(table_name)
  sdk = DynamoDBPartiQLSingle.new(table_name)

  new_step(1, "Create a new DynamoDB table if none already exists.")
  unless scaffold.exists?(table_name)
    puts("\nNo such table: #{table_name}. Creating it...")
    scaffold.create_table(table_name)
    print "Done!\n".green
  end

  new_step(2, "Populate DynamoDB table with movie data.")
  download_file = "moviedata.json"
  puts("Downloading movie database to #{download_file}...")
  movie_data = scaffold.fetch_movie_data(download_file)
  puts("Writing movie data from #{download_file} into your table...")
  scaffold.write_batch(movie_data)
  puts("Records added: #{movie_data.length}.")
  print "Done!\n".green

  new_step(3, "Select a single item from the movies table.")
  response = sdk.select_item_by_title("Star Wars")
  puts("Items selected for title 'Star Wars': #{response.items.length}\n")
  print "#{response.items.first}".yellow
  print "\n\nDone!\n".green

  new_step(4, "Update a single item from the movies table.")
  puts "Let's correct the rating on The Big Lebowski to 10.0."
  sdk.update_rating_by_title("The Big Lebowski", 1998, 10.0)
```

```
  print "\nDone!\n".green

  new_step(5, "Delete a single item from the movies table.")
  puts "Let's delete The Silence of the Lambs because it's just too scary."
  sdk.delete_item_by_title("The Silence of the Lambs", 1991)
  print "\nDone!\n".green

  new_step(6, "Insert a new item into the movies table.")
  puts "Let's create a less-scary movie called The Prancing of the Lambs."
  sdk.insert_item("The Prancing of the Lambs", 2005, "A movie about happy livestock.",
 5.0)
  print "\nDone!\n".green

  new_step(7, "Delete the table.")
  if scaffold.exists?(table_name)
    scaffold.delete_table
  end
end
```

- 有关 API 的详细信息，请参阅 ExecuteStatementAWS SDK for RubyAPI 参考文档。

EventBridge使用 Ruby 软件开发工具包的示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与执行操作和实现常见场景EventBridge。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 场景 (p. 205)

场景

创建和触发规则

以下代码示例显示了如何在 Amazon 中创建和触发规则EventBridge。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

以正确的顺序调用函数。

```
require "aws-sdk-sns"
require "aws-sdk-iam"
require "aws-sdk-cloudwatchevents"
require "aws-sdk-ec2"
require "aws-sdk-cloudwatch"
require "aws-sdk-cloudwatchlogs"
require "securerandom"
```

检查为该函数提供的主题中是否存在指定的亚马逊简单通知服务 (Amazon SNS) 主题。

```
# Checks whether the specified Amazon SNS
# topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end

def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end
```

检查 Amazon SNS 中调用者可用的主题中是否存在指定主题。

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

在 Amazon SNS 中创建一个主题，然后订阅一个电子邮件地址以接收有关该主题的通知。

```
# Creates a topic in Amazon SNS
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The ARN of the topic that was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
    protocol: "email",
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts "Subscription created with ARN " \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \
    "and confirm the subscription to start receiving notification emails."
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return "Error"
end
```

检查提供给此函数的角色中是否存在指定 AWS Identity and Access Management (IAM) 角色。

```
# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end
```

检查 IAM 中调用者可用的角色中是否存在指定的角色。

```
# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts "Role found."
          return true
        end
      end
    end
  end
  puts "Role not found."
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end
```

在 IAM 中创建角色。

```
# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
```

```
            'Service': "events.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }.to_json,
    path: "/",
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts "Adding access policy to role..."
  iam_client.put_role_policy(
    policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "CloudWatchEventsFullAccess",
          'Effect': "Allow",
          'Resource': "*",
          'Action': "events:*"
        },
        {
          'Sid': "IAMPassRoleForCloudWatchEvents",
          'Effect': "Allow",
          'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
          'Action': "iam:PassRole"
        }
      ]
    }.to_json,
    policy_name: "CloudWatchEventsPolicy",
    role_name: role_name
  )
  puts "Access policy added to role."
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end
```

检查提供给此函数的EventBridge规则中是否存在指定的规则。

```
# Checks whether the specified Amazon EventBridge rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end
```

检查调用者可用的规则中是否存在指定的规则EventBridge。

```
# Checks whether the specified rule exists among those available to the
# caller in Amazon EventBridge.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts "Rule found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.rules.count.positive?
        if rule_found?(response.rules, rule_name)
          puts "Rule found."
          return true
        end
      end
    end
  end
  puts "Rule not found."
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end
```

在 EventBridge中创建规则。

```
# Creates a rule in Amazon EventBridge.
# This rule is triggered whenever an available instance in
# Amazon EC2 changes to the specified state.
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon SNS.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon EC2 must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
```

```
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
        "aws.ec2"
      ],
      'detail-type': [
        "EC2 Instance State-change Notification"
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: "ENABLED",
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
      put_targets_response.failed_entry_count > 0
    puts "Error(s) adding target to rule:"
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts "If the rule was created, you must add the target " \
    "to the rule yourself, or delete the rule yourself and try again."
  return false
end
```

在 Amazon Logs 中查看调用者可用的日志组中是否存在指定的CloudWatch日志组。

```
# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts "Log group found."
        return true
      end
    end
  end
  puts "Log group not found."
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
end
```

在 Log CloudWatch s 中创建日志组。

```
# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts "Log group created."
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end
```

在 Logs 中将事件写入CloudWatch日志流。

```
# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts "Message logged."
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end
```

重启亚马逊弹性计算云 (Amazon EC2) 实例，并将有关相关活动的信息添加到 Log CloudWatch s 中的日志流。

```
# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
```

```
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
```

```
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end
```

在中显示有关规则活动的信息EventBridge。

```
# Displays information about activity for a rule in Amazon EventBridge.
#
# Prerequisites:
#
# - A rule in Amazon EventBridge.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts "Attempting to display rule activity..."
  response = cloudwatch_client.get_metric_statistics(
    namespace: "AWS/Events",
    metric_name: "Invocations",
    dimensions: [
      {
        name: "RuleName",
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ["Sum"],
    unit: "Count"
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
```

```
      "specified time period."
  end
rescue StandardError => e
  puts "Error getting information about event rule activity: #{e.message}"
end
```

显示日志组中所有日志流CloudWatch的日志信息。

```
# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts "Attempting to display log stream data for the log group " \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: "LastEventTime",
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
      describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts "-" * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
          get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
        end
      else
        puts "No log messages for this log stream."
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about the log streams or their messages: " \
    "#{e.message}"
end
```

向呼叫者显示提醒，要求他们手动清理他们不再需要的所有关联AWS资源。

```

# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
```

```
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon EventBridge rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts "-" * 10
  puts "Some of the following AWS resources might still exist in your account."
  puts "If you no longer want to use this code example, then to clean up"
  puts "your AWS account and avoid unexpected costs, you might want to"
  puts "manually delete any of the following resources if they exist:"
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
  puts "- The Amazon EventBridge rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Full example call:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = "aws-doc-sdk-examples-topic"
  email_address = "mary@example.com"
  # Properties for the IAM role.
  role_name = "aws-doc-sdk-examples-cloudwatch-events-rule-role"
  # Properties for the Amazon EventBridge rule.
  rule_name = "aws-doc-sdk-examples-ec2-state-change"
  rule_description = "Triggers when any available EC2 instance starts."
  instance_state = "running"
  target_id = "sns-topic"
  # Properties for the Amazon EC2 instance.
  instance_id = "i-033c48ef067af3dEX"
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                              # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = "aws-doc-sdk-examples-cloudwatch-log"
  # AWS service clients for this code example.
  region = "us-east-1"
  sts_client = Aws::STS::Client.new(region: region)
  sns_client = Aws::SNS::Client.new(region: region)
  iam_client = Aws::IAM::Client.new(region: region)
  cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
  ec2_client = Aws::EC2::Client.new(region: region)
  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

  # Get the caller's account ID for use in forming
  # Amazon Resource Names (ARNs) that this code relies on later.
  account_id = sts_client.get_caller_identity.account

  # If the Amazon SNS topic doesn't exist, create it.
  topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
  unless topic_exists?(sns_client, topic_arn)
```

```
  topic_arn = create_topic(sns_client, topic_name, email_address)
  if topic_arn == "Error"
    puts "Could not create the Amazon SNS topic correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam::#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == "Error"
    puts "Could not create the IAM role correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon EventBridge rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts "Could not create the Amazon EventBridge rule correctly. " \
      "Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Could not create the Amazon CloudWatch Logs log group " \
    "correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# Restart the Amazon EC2 instance, which triggers the rule.
unless instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  puts "Could not restart the instance to trigger the rule. " \
    "Continuing anyway to show information about the rule and logs..."
end

# Display how many times the rule was triggered over the past 10 minutes.
display_rule_activity(
  cloudwatch_client,
```

```
    rule_name,
    start_time,
    end_time,
    period
  )

  # Display related log data in Amazon CloudWatch Logs.
  display_log_data(cloudwatchlogs_client, log_group_name)

  # Reminder the caller to clean up any AWS resources that are used
  # by this code example and are no longer needed.
  manual_cleanup_notice(
    topic_name, role_name, rule_name, log_group_name, instance_id
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - PutEvents
  - PutRule

AWS Glue使用 Ruby 软件开发工具包的示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与执行操作和实现常见场景AWS Glue。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)
- 场景 (p. 205)

操作

创建爬网程序

以下代码示例显示了如何创建 AWS Glue Crawler。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
```

```
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new crawler with the specified configuration.
  #
  # @param name [String] The name of the crawler.
  # @param role_arn [String] The ARN of the IAM role to be used by the crawler.
  # @param db_name [String] The name of the database where the crawler stores its
 metadata.
  # @param db_prefix [String] The prefix to be added to the names of tables that the
 crawler creates.
  # @param s3_target [String] The S3 path that the crawler will crawl.
  # @return [void]
  def create_crawler(name, role_arn, db_name, db_prefix, s3_target)
    @glue_client.create_crawler(
      name: name,
      role: role_arn,
      database_name: db_name,
      targets: {
        s3_targets: [
          {
            path: s3_target
          }
        ]
      }
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create crawler: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 CreateCrawlerAWS SDK for RubyAPI 参考文档。

创建作业定义

以下代码示例显示了如何创建AWS Glue作业定义。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new job with the specified configuration.
  #
```

```
  # @param name [String] The name of the job.
  # @param description [String] The description of the job.
  # @param role_arn [String] The ARN of the IAM role to be used by the job.
  # @param script_location [String] The location of the ETL script for the job.
  # @return [void]
  def create_job(name, description, role_arn, script_location)
    @glue_client.create_job(
      name: name,
      description: description,
      role: role_arn,
      command: {
        name: "glueetl",
        script_location: script_location,
        python_version: "3"
      },
      glue_version: "3.0"
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create job #{name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 CreateJobAWS SDK for RubyAPI 参考文档。

删除爬网程序

以下代码示例显示了如何删除 AWS Glue Crawler。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to delete.
  # @return [void]
  def delete_crawler(name)
    @glue_client.delete_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DeleteCrawlerAWS SDK for RubyAPI 参考文档。

从 Data Catalog 中删除数据库

以下代码示例显示了如何从中删除数据库AWS Glue Data Catalog。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Removes a specified database from a Data Catalog.
  #
  # @param database_name [String] The name of the database to delete.
  # @return [void]
  def delete_database(database_name)
    @glue_client.delete_database(name: database_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete database: \n#{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 DeleteDatabaseAWS SDK for RubyAPI 参考文档。

删除作业定义

以下代码示例显示了如何删除AWS Glue作业定义和所有关联的运行。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end
```

```
  # Deletes a job with the specified name.
  #
  # @param job_name [String] The name of the job to delete.
  # @return [void]
  def delete_job(job_name)
    @glue_client.delete_job(job_name: job_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 DeleteJobAWS SDK for RubyAPI 参考文档。

从数据库中删除表

以下代码示例显示了如何从AWS Glue Data Catalog数据库中删除表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a table with the specified name.
  #
  # @param database_name [String] The name of the catalog database in which the table
 resides.
  # @param table_name [String] The name of the table to be deleted.
  # @return [void]
  def delete_table(database_name, table_name)
    @glue_client.delete_table(database_name: database_name, name: table_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 DeleteTableAWS SDK for RubyAPI 参考文档。

获取爬网程序

以下代码示例显示了如何获取AWS Glue爬虫。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
  # @param name [String] The name of the crawler to retrieve information about.
  # @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil if not
 found.
  def get_crawler(name)
    @glue_client.get_crawler(name: name)
  rescue Aws::Glue::Errors::EntityNotFoundException
    @logger.info("Crawler #{name} doesn't exist.")
    false
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 GetCrawlerAWS SDK for RubyAPI 参考文档。

从 Data Catalog 获取数据库

以下代码示例显示了如何从中获取数据库AWS Glue Data Catalog。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific database.
  #
  # @param name [String] The name of the database to retrieve information about.
  # @return [Aws::Glue::Types::Database, nil] The database object if found, or nil if
 not found.
  def get_database(name)
```

```
    response = @glue_client.get_database(name: name)
    response.database
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get database #{name}: \n#{e.message}")
  raise
  end
```

- 有关 API 的详细信息，请参阅 GetDatabaseAWS SDK for RubyAPI 参考文档。

让作业运行

以下代码示例显示了如何运行作AWS Glue业。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves data for a specific job run.
  #
  # @param job_name [String] The name of the job run to retrieve data for.
  # @return [Glue::Types::GetJobRunResponse]
  def get_job_run(job_name, run_id)
    @glue_client.get_job_run(job_name: job_name, run_id: run_id)
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 GetJobRunAWS SDK for RubyAPI 参考文档。

获取作业运行情况

以下代码示例显示了如何运行作AWS Glue业。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
```

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of job runs for the specified job.
  #
  # @param job_name [String] The name of the job to retrieve job runs for.
  # @return [Array<Aws::Glue::Types::JobRun>]
  def get_job_runs(job_name)
    response = @glue_client.get_job_runs(job_name: job_name)
    response.job_runs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 GetJobRunsAWS SDK for RubyAPI 参考文档。

从数据库获取表

以下代码示例显示了如何从中的数据库获取表AWS Glue Data Catalog。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of tables in the specified database.
  #
  # @param db_name [String] The name of the database to retrieve tables from.
  # @return [Array<Aws::Glue::Types::Table>]
  def get_tables(db_name)
    response = @glue_client.get_tables(database_name: db_name)
    response.table_list
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 GetTablesAWS SDK for RubyAPI 参考文档。

列出作业定义。

以下代码示例显示了如何列出AWS Glue作业定义。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of jobs in AWS Glue.
  #
  # @return [Aws::Glue::Types::ListJobsResponse]
  def list_jobs
    @glue_client.list_jobs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not list jobs: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 ListJobsAWS SDK for RubyAPI 参考文档。

启动爬网程序

以下代码示例显示了如何启动 AWS Glue Crawler。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
```

```
    @glue_client = glue_client
    @logger = logger
  end

  # Starts a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to start.
  # @return [void]
  def start_crawler(name)
    @glue_client.start_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 StartCrawlerAWS SDK for RubyAPI 参考文档。

启动任务运行

以下代码示例显示了如何开始运行作AWS Glue业。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Starts a job run for the specified job.
  #
  # @param name [String] The name of the job to start the run for.
  # @param input_database [String] The name of the input database for the job.
  # @param input_table [String] The name of the input table for the job.
  # @param output_bucket_name [String] The name of the output S3 bucket for the job.
  # @return [String] The ID of the started job run.
  def start_job_run(name, input_database, input_table, output_bucket_name)
    response = @glue_client.start_job_run(
      job_name: name,
      arguments: {
        '--input_database': input_database,
        '--input_table': input_table,
        '--output_bucket_url': "s3://#{output_bucket_name}/"
      }
    )
    response.job_run_id
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not start job run #{name}: \n#{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 StartJobRunAWS SDK for RubyAPI 参考文档。

场景

开始使用爬虫和作业

以下代码示例显示了如何：

- 创建一个搜寻器，用于抓取公共 Amazon S3 存储桶并生成 CSV 格式的元数据数据库。
- 在 AWS Glue Data Catalog 中列出有关数据库和表的信息。
- 创建任务以从 S3 存储桶提取 CSV 数据，转换数据，然后将 JSON 格式的输出加载到另一个 S3 存储桶中。
- 列出有关作业运行的信息，查看转换后的数据和清理资源。

有关更多信息，请参阅教程：AWS Glue Studio 入门。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

创建一个包含场景中使用的 AWS Glue 函数的类。

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing a
 simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods for
 interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API calls
 and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
  # @param name [String] The name of the crawler to retrieve information about.
  # @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil if not
 found.
  def get_crawler(name)
    @glue_client.get_crawler(name: name)
  rescue Aws::Glue::Errors::EntityNotFoundException
    @logger.info("Crawler #{name} doesn't exist.")
    false
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
    raise
  end

  # Creates a new crawler with the specified configuration.
  #
  # @param name [String] The name of the crawler.
  # @param role_arn [String] The ARN of the IAM role to be used by the crawler.
  # @param db_name [String] The name of the database where the crawler stores its
 metadata.
```

```
  # @param db_prefix [String] The prefix to be added to the names of tables that the
 crawler creates.
  # @param s3_target [String] The S3 path that the crawler will crawl.
  # @return [void]
  def create_crawler(name, role_arn, db_name, db_prefix, s3_target)
    @glue_client.create_crawler(
      name: name,
      role: role_arn,
      database_name: db_name,
      targets: {
        s3_targets: [
          {
            path: s3_target
          }
        ]
      }
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create crawler: \n#{e.message}")
    raise
  end

  # Starts a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to start.
  # @return [void]
  def start_crawler(name)
    @glue_client.start_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
    raise
  end

  # Deletes a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to delete.
  # @return [void]
  def delete_crawler(name)
    @glue_client.delete_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
    raise
  end

  # Retrieves information about a specific database.
  #
  # @param name [String] The name of the database to retrieve information about.
  # @return [Aws::Glue::Types::Database, nil] The database object if found, or nil if
 not found.
  def get_database(name)
    response = @glue_client.get_database(name: name)
    response.database
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get database #{name}: \n#{e.message}")
  raise
  end

  # Retrieves a list of tables in the specified database.
  #
  # @param db_name [String] The name of the database to retrieve tables from.
  # @return [Array<Aws::Glue::Types::Table>]
  def get_tables(db_name)
    response = @glue_client.get_tables(database_name: db_name)
    response.table_list
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
```

```
  raise
end

# Creates a new job with the specified configuration.
#
# @param name [String] The name of the job.
# @param description [String] The description of the job.
# @param role_arn [String] The ARN of the IAM role to be used by the job.
# @param script_location [String] The location of the ETL script for the job.
# @return [void]
def create_job(name, description, role_arn, script_location)
  @glue_client.create_job(
    name: name,
    description: description,
    role: role_arn,
    command: {
      name: "glueetl",
      script_location: script_location,
      python_version: "3"
    },
    glue_version: "3.0"
  )
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create job #{name}: \n#{e.message}")
  raise
end

# Starts a job run for the specified job.
#
# @param name [String] The name of the job to start the run for.
# @param input_database [String] The name of the input database for the job.
# @param input_table [String] The name of the input table for the job.
# @param output_bucket_name [String] The name of the output S3 bucket for the job.
# @return [String] The ID of the started job run.
def start_job_run(name, input_database, input_table, output_bucket_name)
  response = @glue_client.start_job_run(
    job_name: name,
    arguments: {
      '--input_database': input_database,
      '--input_table': input_table,
      '--output_bucket_url': "s3://#{output_bucket_name}/"
    }
  )
  response.job_run_id
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not start job run #{name}: \n#{e.message}")
  raise
end

# Retrieves a list of jobs in AWS Glue.
#
# @return [Aws::Glue::Types::ListJobsResponse]
def list_jobs
  @glue_client.list_jobs
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not list jobs: \n#{e.message}")
  raise
end

# Retrieves a list of job runs for the specified job.
#
# @param job_name [String] The name of the job to retrieve job runs for.
# @return [Array<Aws::Glue::Types::JobRun>]
def get_job_runs(job_name)
  response = @glue_client.get_job_runs(job_name: job_name)
  response.job_runs
```

```
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end

  # Retrieves data for a specific job run.
  #
  # @param job_name [String] The name of the job run to retrieve data for.
  # @return [Glue::Types::GetJobRunResponse]
  def get_job_run(job_name, run_id)
    @glue_client.get_job_run(job_name: job_name, run_id: run_id)
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end

  # Deletes a job with the specified name.
  #
  # @param job_name [String] The name of the job to delete.
  # @return [void]
  def delete_job(job_name)
    @glue_client.delete_job(job_name: job_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end

  # Deletes a table with the specified name.
  #
  # @param database_name [String] The name of the catalog database in which the table
 resides.
  # @param table_name [String] The name of the table to be deleted.
  # @return [void]
  def delete_table(database_name, table_name)
    @glue_client.delete_table(database_name: database_name, name: table_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end

  # Removes a specified database from a Data Catalog.
  #
  # @param database_name [String] The name of the database to delete.
  # @return [void]
  def delete_database(database_name)
    @glue_client.delete_database(name: database_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete database: \n#{e.message}")
  end

  # Uploads a job script file to an S3 bucket.
  #
  # @param file_path [String] The local path of the job script file.
  # @param bucket_resource [Aws::S3::Bucket] The S3 bucket resource to upload the file
 to.
  # @return [void]
  def upload_job_script(file_path, bucket_resource)
    File.open(file_path) do |file|
      bucket_resource.client.put_object({
       body: file,
       bucket: bucket_resource.name,
       key: file_path
     })
    end
  rescue Aws::S3::Errors::S3UploadFailedError => e
    @logger.error("S3 could not upload job script: \n#{e.message}")
    raise
  end

end
```

创建运行场景的类。

```
class GlueCrawlerJobScenario
  def initialize(glue_client, glue_service_role, glue_bucket, logger)
    @glue_client = glue_client
    @glue_service_role = glue_service_role
    @glue_bucket = glue_bucket
    @logger = logger
  end

  def run(crawler_name, db_name, db_prefix, data_source, job_script, job_name)
    wrapper = GlueWrapper.new(@glue_client, @logger)

    new_step(1, "Create a crawler")
    puts "Checking for crawler #{crawler_name}."
    crawler = wrapper.get_crawler(crawler_name)
    if crawler == false
      puts "Creating crawler #{crawler_name}."
      wrapper.create_crawler(crawler_name, @glue_service_role.arn, db_name, db_prefix,
 data_source)
      puts "Successfully created #{crawler_name}:"
      crawler = wrapper.get_crawler(crawler_name)
      puts JSON.pretty_generate(crawler).yellow
    end
    print "\nDone!\n".green

    new_step(2, "Run a crawler to output a database.")
    puts "Location of input data analyzed by crawler: #{data_source}"
    puts "Outputs: a Data Catalog database in CSV format containing metadata on input."
    wrapper.start_crawler(crawler_name)
    puts "Starting crawler... (this typically takes a few minutes)"
    crawler_state = nil
    while crawler_state != "READY"
      custom_wait(15)
      crawler = wrapper.get_crawler(crawler_name)
      crawler_state = crawler[0]["state"]
      print "Status check: #{crawler_state}.".yellow
    end
    print "\nDone!\n".green

    new_step(3, "Query the database.")
    database = wrapper.get_database(db_name)
    puts "The crawler created database #{db_name}:"
    print "#{database}".yellow
    puts "\nThe database contains these tables:"
    tables = wrapper.get_tables(db_name)
    tables.each_with_index do |table, index|
      print "\t#{index + 1}. #{table['name']}".yellow
    end
    print "\nDone!\n".green

    new_step(4, "Create a job definition that runs an ETL script.")
    puts "Uploading Python ETL script to S3..."
    wrapper.upload_job_script(job_script, @glue_bucket)
    puts "Creating job definition #{job_name}:\n"
    response = wrapper.create_job(job_name, "Getting started example job.",
 @glue_service_role.arn, "s3://#{@glue_bucket.name}/#{job_script}")
    puts JSON.pretty_generate(response).yellow
    print "\nDone!\n".green

    new_step(5, "Start a new job")
    job_run_status = nil
    job_run_id = wrapper.start_job_run(
```

```
      job_name,
      db_name,
      tables[0]["name"],
      @glue_bucket.name
    )
    puts "Job #{job_name} started. Let's wait for it to run."
    until ["SUCCEEDED", "STOPPED", "FAILED", "TIMEOUT"].include?(job_run_status)
      custom_wait(10)
      job_run = wrapper.get_job_runs(job_name)
      job_run_status = job_run[0]["job_run_state"]
      print "Status check: #{job_name}/#{job_run_id} - #{job_run_status}.".yellow
    end
    print "\nDone!\n".green

    new_step(6, "View results from a successful job run.")
    if job_run_status == "SUCCEEDED"
      puts "Data from your job run is stored in your S3 bucket '#{@glue_bucket.name}'.
 Files include:"
      begin

        # Print the key name of each object in the bucket.
        @glue_bucket.objects.each do |object_summary|
          if object_summary.key.include?("run-")
            print "#{object_summary.key}".yellow
          end
        end

        # Print the first 256 bytes of a run file
        desired_sample_objects = 1
        @glue_bucket.objects.each do |object_summary|
          if object_summary.key.include?("run-")
            if desired_sample_objects > 0
              sample_object = @glue_bucket.object(object_summary.key)
              sample = sample_object.get(range: "bytes=0-255").body.read
              puts "\nSample run file contents:"
              print "#{sample}".yellow
              desired_sample_objects -= 1
            end
          end
        end
      rescue Aws::S3::Errors::ServiceError => e
        logger.error(
          "Couldn't get job run data. Here's why: %s: %s",
          e.response.error.code, e.response.error.message
        )
        raise
      end
    end
    print "\nDone!\n".green

    new_step(7, "Delete job definition and crawler.")
    wrapper.delete_job(job_name)
    puts "Job deleted: #{job_name}."
    wrapper.delete_crawler(crawler_name)
    puts "Crawler deleted: #{crawler_name}."
    wrapper.delete_table(db_name, tables[0]["name"])
    puts "Table deleted: #{tables[0]["name"]} in #{db_name}."
    wrapper.delete_database(db_name)
    puts "Database deleted: #{db_name}."
    print "\nDone!\n".green
  end
end

def main

  banner("../../helpers/banner.txt")
```

```
 puts
"#################################################################################################
 puts "#
                     #".yellow
 puts "#                                              EXAMPLE CODE DEMO:
                     #".yellow
 puts "#                                                  AWS Glue
                     #".yellow
 puts "#
                     #".yellow
 puts
"#################################################################################################
 puts ""
 puts "You have launched a demo of AWS Glue using the AWS for Ruby v3 SDK. Over the
next 60 seconds, it will"
 puts "do the following:"
 puts "    1. Create a crawler."
 puts "    2. Run a crawler to output a database."
 puts "    3. Query the database."
 puts "    4. Create a job definition that runs an ETL script."
 puts "    5. Start a new job."
 puts "    6. View results from a successful job run."
 puts "    7. Delete job definition and crawler."
 puts ""

 confirm_begin
 billing
 security
 puts "\e[H\e[2J"

 # Set input file names
 job_script_filepath = "job_script.py"
 resource_names = YAML.load_file("resource_names.yaml")

 # Instantiate existing IAM role.
 iam = Aws::IAM::Resource.new(region: "us-east-1")
 iam_role_name = resource_names["glue_service_role"]
 iam_role = iam.role(iam_role_name)

 # Instantiate existing S3 bucket.
 s3 = Aws::S3::Resource.new(region: "us-east-1")
 s3_bucket_name = resource_names["glue_bucket"]
 s3_bucket = s3.bucket(s3_bucket_name)

 scenario = GlueCrawlerJobScenario.new(
   Aws::Glue::Client.new(region: "us-east-1"),
   iam_role,
   s3_bucket,
   @logger
 )

 random_int = rand(10 ** 4)
 scenario.run(
   "doc-example-crawler-#{random_int}",
   "doc-example-database-#{random_int}",
   "doc-example-#{random_int}-",
   "s3://crawler-public-us-east-1/flight/2016/csv",
   job_script_filepath,
   "doc-example-job-#{random_int}"
 )

 puts "-" * 88
 puts "You have reached the end of this tour of AWS Glue."
 puts "To destroy CDK-created resources, run:\n       cdk destroy"
 puts "-" * 88
```

```
end
```

创建一个 ETL 脚本，AWS Glue 用此脚本在作业运行期间提取、转换和加载数据。

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

"""
These custom arguments must be passed as Arguments to the StartJobRun request.
    --input_database    The name of a metadata database that is contained in your
                        AWS Glue Data Catalog and that contains tables that describe
                        the data to be processed.
    --input_table       The name of a table in the database that describes the data to
                        be processed.
    --output_bucket_url An S3 bucket that receives the transformed output data.
"""
args = getResolvedOptions(sys.argv, [
    "JOB_NAME", "input_database", "input_table", "output_bucket_url"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

# Script generated for node S3 Flight Data.
S3FlightData_node1 = glueContext.create_dynamic_frame.from_catalog(
    database=args['input_database'],
    table_name=args['input_table'],
    transformation_ctx="S3FlightData_node1",
)

# This mapping performs two main functions:
# 1. It simplifies the output by removing most of the fields from the data.
# 2. It renames some fields. For example, `fl_date` is renamed to `flight_date`.
ApplyMapping_node2 = ApplyMapping.apply(
    frame=S3FlightData_node1,
    mappings=[
        ("year", "long", "year", "long"),
        ("month", "long", "month", "tinyint"),
        ("day_of_month", "long", "day", "tinyint"),
        ("fl_date", "string", "flight_date", "string"),
        ("carrier", "string", "carrier", "string"),
        ("fl_num", "long", "flight_num", "long"),
        ("origin_city_name", "string", "origin_city_name", "string"),
        ("origin_state_abr", "string", "origin_state_abr", "string"),
        ("dest_city_name", "string", "dest_city_name", "string"),
        ("dest_state_abr", "string", "dest_state_abr", "string"),
        ("dep_time", "long", "departure_time", "long"),
        ("wheels_off", "long", "wheels_off", "long"),
        ("wheels_on", "long", "wheels_on", "long"),
        ("arr_time", "long", "arrival_time", "long"),
        ("mon", "string", "mon", "string"),
    ],
    transformation_ctx="ApplyMapping_node2",
)

# Script generated for node Revised Flight Data.
RevisedFlightData_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
```

```
    format="json",
    connection_options={"path": args['output_bucket_url'], "partitionKeys": []},
    transformation_ctx="RevisedFlightData_node3",
)

job.commit()
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

使用 Ruby 开发工具包的 IAM 示例

以下代码示例向您展示了如何使用 with IAM 来执行操作和实现常见场景。AWS SDK for Ruby

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)
- 场景 (p. 205)

操作

将策略附加到角色

以下代码示例显示了如何将 IAM 策略附加到角色。

SDK for Ruby

> Note
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates a policy that grants permission to list S3 buckets in the account, and
# then attaches the policy to a role.
#
# @param policy_name [String] The name to give the policy.
# @param role [Aws::IAM::Role] The role that the policy is attached to.
# @return [Aws::IAM::Policy] The newly created policy.
def create_and_attach_role_policy(policy_name, role)
  policy = @iam_resource.create_policy(
    policy_name: policy_name,
    policy_document: {
      Version: "2012-10-17",
      Statement: [{
        Effect: "Allow",
        Action: "s3:ListAllMyBuckets",
        Resource: "arn:aws:s3:::*"
      }]
    }.to_json)
  role.attach_policy(policy_arn: policy.arn)
  puts("Created policy #{policy.policy_name} and attached it to role #{role.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't create a policy and attach it to role #{role.name}. Here's why: ")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  policy
end
```

- 有关 API 的详细信息，请参阅 AttachRolePolicyAWS SDK for RubyAPI 参考文档。

创建策略

以下代码示例显示了如何创建 IAM 策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates a policy that grants permission to list S3 buckets in the account, and
# then attaches the policy to a role.
#
# @param policy_name [String] The name to give the policy.
# @param role [Aws::IAM::Role] The role that the policy is attached to.
# @return [Aws::IAM::Policy] The newly created policy.
def create_and_attach_role_policy(policy_name, role)
  policy = @iam_resource.create_policy(
    policy_name: policy_name,
    policy_document: {
      Version: "2012-10-17",
      Statement: [{
        Effect: "Allow",
        Action: "s3:ListAllMyBuckets",
        Resource: "arn:aws:s3:::*"
      }]
    }.to_json)
  role.attach_policy(policy_arn: policy.arn)
  puts("Created policy #{policy.policy_name} and attached it to role #{role.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't create a policy and attach it to role #{role.name}. Here's why: ")
  puts("\t#{e.code}: #{e.message}")
```

```
  raise
else
  policy
end
```

- 有关 API 的详细信息，请参阅 CreatePolicyAWS SDK for RubyAPI 参考文档。

创建角色

以下代码示例显示了如何创建 IAM 角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates a role that can be assumed by a user.
#
# @param role_name [String] The name to give the role.
# @param user [Aws::IAM::User] The user who is granted permission to assume the role.
# @return [Aws::IAM::Role] The newly created role.
def create_role(role_name, user)
  role = @iam_resource.create_role(
    role_name: role_name,
    assume_role_policy_document: {
      Version: "2012-10-17",
      Statement: [{
        Effect: "Allow",
        Principal: {'AWS': user.arn},
        Action: "sts:AssumeRole"
      }]
    }.to_json)
  puts("Created role #{role.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't create a role for the demo. Here's why: ")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
end
```

- 有关 API 的详细信息，请参阅 CreateRoleAWS SDK for RubyAPI 参考文档。

创建服务相关角色

以下代码示例显示了如何创建 IAM 服务相关角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates a service-linked role.
#
# @param service_name [String] The name of the service that owns the role.
# @param description [String] A description to give the role.
```

```
# @return [Aws::IAM::Role] The newly created role.
def create_service_linked_role(service_name, description)
  response = @iam_resource.client.create_service_linked_role(
    aws_service_name: service_name, description: description)
  role = @iam_resource.role(response.role.role_name)
  puts("Created service-linked role #{role.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't create service-linked role for #{service_name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
end
```

- 有关 API 的详细信息，请参阅 CreateServiceLinkedRoleAWS SDK for RubyAPI 参考文档。

创建用户

以下代码示例显示了如何创建 IAM 用户。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates a user.
#
# @param user_name [String] The name to give the user.
# @return [Aws::IAM::User] The newly created user.
def create_user(user_name)
  user = @iam_resource.create_user(user_name: user_name)
  puts("Created demo user named #{user.name}.")
rescue Aws::Errors::ServiceError => e
  puts("Tried and failed to create demo user.")
  puts("\t#{e.code}: #{e.message}")
  puts("\nCan't continue the demo without a user!")
  raise
else
  user
end
```

- 有关 API 的详细信息，请参阅 CreateUserAWS SDK for RubyAPI 参考文档。

创建访问密钥

以下代码示例显示了如何创建 IAM 访问密钥。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates an access key for a user.
#
# @param user [Aws::IAM::User] The user that owns the key.
# @return [Aws::IAM::AccessKeyPair] The newly created access key.
def create_access_key_pair(user)
  user_key = user.create_access_key_pair
  puts("Created access key pair for user.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't create access keys for user #{user.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  user_key
end
```

- 有关 API 的详细信息，请参阅 CreateAccessKeyAWS SDK for RubyAPI 参考文档。

为用户创建内联策略

以下代码示例演示如何为用户创建内联 IAM policy。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Creates an inline policy for a user that lets the user assume a role.
#
# @param policy_name [String] The name to give the policy.
# @param user [Aws::IAM::User] The user that owns the policy.
# @param role [Aws::IAM::Role] The role that can be assumed.
# @return [Aws::IAM::UserPolicy] The newly created policy.
def create_user_policy(policy_name, user, role)
  policy = user.create_policy(
    policy_name: policy_name,
    policy_document: {
      Version: "2012-10-17",
      Statement: [{
        Effect: "Allow",
        Action: "sts:AssumeRole",
        Resource: role.arn
      }]
    }.to_json)
  puts("Created an inline policy for #{user.name} that lets the user assume role
#{role.name}.")
 rescue Aws::Errors::ServiceError => e
```

```
  puts("Couldn't create an inline policy for user #{user.name}. Here's why: ")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  policy
end
```

- 有关 API 的详细信息，请参阅 PutUserPolicyAWS SDK for RubyAPI 参考文档。

删除策略

以下代码示例显示了如何删除 IAM 策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a role. If the role has policies attached, they are detached and
# deleted before the role is deleted.
#
# @param role [Aws::IAM::Role] The role to delete.
def delete_role(role)
  role.attached_policies.each do |policy|
    name = policy.policy_name
    policy.detach_role(role_name: role.name)
    policy.delete
    puts("Deleted policy #{name}.")
  end
  name = role.name
  role.delete
  puts("Deleted role #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete role #{role.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 DeletePolicyAWS SDK for RubyAPI 参考文档。

删除角色

以下代码示例显示了如何删除 IAM 角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a role. If the role has policies attached, they are detached and
# deleted before the role is deleted.
#
# @param role [Aws::IAM::Role] The role to delete.
def delete_role(role)
  role.attached_policies.each do |policy|
```

```
      name = policy.policy_name
      policy.detach_role(role_name: role.name)
      policy.delete
      puts("Deleted policy #{name}.")
    end
    name = role.name
    role.delete
    puts("Deleted role #{name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't detach policies and delete role #{role.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DeleteRoleAWS SDK for RubyAPI 参考文档。

删除服务相关角色

以下代码示例显示了如何删除 IAM 服务相关角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
  # Deletes a service-linked role from the account.
  #
  # @param role [Aws::IAM::Role] The role to delete.
  def delete_service_linked_role(role)
    response = @iam_resource.client.delete_service_linked_role(role_name: role.name)
    task_id = response.deletion_task_id
    while true
      response = @iam_resource.client.get_service_linked_role_deletion_status(
        deletion_task_id: task_id)
      status = response.status
      puts("Deletion of #{role.name} #{status}.")
      if %w(SUCCEEDED FAILED).include?(status)
        break
      else
        sleep(3)
      end
    end
rescue Aws::Errors::ServiceError => e
  # If AWS has not yet fully propagated the role, it deletes the role but
  # returns NoSuchEntity.
  if e.code != "NoSuchEntity"
    puts("Couldn't delete #{role.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
  end
```

- 有关 API 的详细信息，请参阅 DeleteServiceLinkedRoleAWS SDK for RubyAPI 参考文档。

删除用户

以下代码示例显示了如何删除 IAM 用户。

Warning

为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a user. If the user has inline policies or access keys, they are deleted
# before the user is deleted.
#
# @param user [Aws::IAM::User] The user to delete.
def delete_user(user)
  user.policies.each do |policy|
    name = policy.name
    policy.delete
    puts("Deleted user policy #{name}.")
  end
  user.access_keys.each do |key|
    key.delete
    puts("Deleted access key for user #{user.name}.")
  end
  name = user.name
  user.delete
  puts("Deleted user #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete user #{user.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
end
```

- 有关 API 的详细信息，请参阅 DeleteUserAWS SDK for RubyAPI 参考文档。

删除访问密钥

以下代码示例显示了如何删除 IAM 访问密钥。

Warning

为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a user. If the user has inline policies or access keys, they are deleted
# before the user is deleted.
#
# @param user [Aws::IAM::User] The user to delete.
def delete_user(user)
```

```
  user.policies.each do |policy|
    name = policy.name
    policy.delete
    puts("Deleted user policy #{name}.")
  end
  user.access_keys.each do |key|
    key.delete
    puts("Deleted access key for user #{user.name}.")
  end
  name = user.name
  user.delete
  puts("Deleted user #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete user #{user.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
end
```

- 有关 API 的详细信息，请参阅 DeleteAccessKeyAWS SDK for RubyAPI 参考文档。

删除用户的内联策略

以下代码示例显示了如何从用户那里删除内联 IAM 策略。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a user. If the user has inline policies or access keys, they are deleted
# before the user is deleted.
#
# @param user [Aws::IAM::User] The user to delete.
def delete_user(user)
  user.policies.each do |policy|
    name = policy.name
    policy.delete
    puts("Deleted user policy #{name}.")
  end
  user.access_keys.each do |key|
    key.delete
    puts("Deleted access key for user #{user.name}.")
  end
  name = user.name
  user.delete
  puts("Deleted user #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete user #{user.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
end
```

- 有关 API 的详细信息，请参阅 DeleteUserPolicyAWS SDK for RubyAPI 参考文档。

从角色分离策略

以下代码示例显示了如何将 IAM 策略与角色分离。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a role. If the role has policies attached, they are detached and
# deleted before the role is deleted.
#
# @param role [Aws::IAM::Role] The role to delete.
def delete_role(role)
  role.attached_policies.each do |policy|
    name = policy.policy_name
    policy.detach_role(role_name: role.name)
    policy.delete
    puts("Deleted policy #{name}.")
  end
  name = role.name
  role.delete
  puts("Deleted role #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete role #{role.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 DetachRolePolicyAWS SDK for RubyAPI 参考文档。

获取策略

以下代码示例显示了如何获取 IAM 策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Gets data about a policy.
#
# @param policy_arn [String] The ARN of the policy to look up.
# @return [Aws::IAM::Policy] The retrieved policy.
def get_policy(policy_arn)
  policy = @iam_resource.policy(policy_arn)
  puts("Got policy '#{policy.policy_name}'. Its ID is: #{policy.policy_id}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't get policy '#{policy_arn}'. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  policy
end
```

- 有关 API 的详细信息，请参阅 GetPolicyAWS SDK for RubyAPI 参考文档。

获取角色

以下代码示例显示了如何获取 IAM 角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Gets data about a role.
#
# @param name [String] The name of the role to look up.
# @return [Aws::IAM::Role] The retrieved role.
def get_role(name)
  role = @iam_resource.role(name)
  puts("Got data for role '#{role.name}'. Its ARN is '#{role.arn}'.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't get data for role '#{name}' Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  role
end
```

- 有关 API 的详细信息，请参阅 GetRoleAWS SDK for RubyAPI 参考文档。

获取账户密码策略

以下代码示例显示了如何获取 IAM 账户密码策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Prints the password policy for the account.
def print_account_password_policy
  policy = @iam_resource.account_password_policy
  policy.load
  puts("The account password policy is:")
  puts(policy.data.to_h)
rescue Aws::Errors::ServiceError => e
  if e.code == "NoSuchEntity"
    puts("The account does not have a password policy.")
  else
    puts("Couldn't print the account password policy. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
end
```

- 有关 API 的详细信息，请参阅 GetAccountPasswordPolicyAWS SDK for RubyAPI 参考文档。

列出 SAML 提供商

以下代码示例显示如何列出 IAM 的 SAML 提供商。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Lists up to a specified number of SAML providers for the account.
#
# @param count [Integer] The maximum number of providers to list.
def list_saml_providers(count)
  @iam_resource.saml_providers.limit(count).each do |provider|
    puts("\t#{provider.arn}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list SAML providers. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 ListSAMLProviders。

列出用户的访问密钥

以下代码示例显示如何列出用户的 IAM 访问密钥。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a user. If the user has inline policies or access keys, they are deleted
# before the user is deleted.
#
# @param user [Aws::IAM::User] The user to delete.
def delete_user(user)
  user.policies.each do |policy|
    name = policy.name
    policy.delete
    puts("Deleted user policy #{name}.")
  end
  user.access_keys.each do |key|
    key.delete
    puts("Deleted access key for user #{user.name}.")
  end
  name = user.name
  user.delete
  puts("Deleted user #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete user #{user.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
end
```

- 有关 API 的详细信息，请参阅 ListAccessKeysAWS SDK for RubyAPI 参考文档。

列出组

以下代码示例显示了如何列出 IAM 群组。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Lists up to a specified number of groups for the account.
#
# @param count [Integer] The maximum number of groups to list.
def list_groups(count)
  @iam_resource.groups.limit(count).each do |group|
    puts("\t#{group.name}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list groups for the account. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 ListGroupsAWS SDK for RubyAPI 参考文档。

列出策略

以下代码示例显示了如何列出 IAM 策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Lists up to a specified number of policies in the account.
#
# @param count [Integer] The maximum number of policies to list.
# @return [Array] The Amazon Resource Names (ARNs) of the policies listed.
def list_policies(count)
  policy_arns = []
  @iam_resource.policies.limit(count).each_with_index do |policy, index|
    puts("\t#{index + 1}: #{policy.policy_name}: #{policy.arn}")
    policy_arns.append(policy.arn)
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list policies for the account. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
else
  policy_arns
end
```

- 有关 API 的详细信息，请参阅 ListPoliciesAWS SDK for RubyAPI 参考文档。

列出附加到角色的策略

以下代码示例显示如何列出附加到 IAM 角色的策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes a role. If the role has policies attached, they are detached and
# deleted before the role is deleted.
#
# @param role [Aws::IAM::Role] The role to delete.
def delete_role(role)
  role.attached_policies.each do |policy|
    name = policy.policy_name
    policy.detach_role(role_name: role.name)
    policy.delete
    puts("Deleted policy #{name}.")
  end
  name = role.name
  role.delete
  puts("Deleted role #{name}.")
rescue Aws::Errors::ServiceError => e
  puts("Couldn't detach policies and delete role #{role.name}. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 ListAttachedRolePoliciesAWS SDK for RubyAPI 参考文档。

列出角色

以下代码示例显示了如何列出 IAM 角色。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Lists up to a specified number of roles for the account.
#
# @param count [Integer] The maximum number of roles to list.
# @return [Array] The names of the listed roles.
def list_roles(count)
  role_names = []
  @iam_resource.roles.limit(count).each_with_index do |role, index|
    puts("\t#{index + 1}: #{role.name}")
    role_names.append(role.name)
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list roles for the account. Here's why:")
  puts("\t#{e.code}: #{e.message}")
```

```
  raise
else
  role_names
end
```

- 有关 API 的详细信息，请参阅 ListRolesAWS SDK for RubyAPI 参考文档。

列出用户

以下代码示例显示了如何列出 IAM 用户。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Lists up to a specified number of users in the account.
#
# @param count [Integer] The maximum number of users to list.
def list_users(count)
  @iam_resource.users.limit(count).each do |user|
    puts("\t#{user.name}")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't list users for the account. Here's why:")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 ListUsersAWS SDK for RubyAPI 参考文档。

场景

创建用户并代入角色

以下代码示例显示了如何创建用户和代入角色。

> **Warning**
>
> 为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 AWS IAM Identity Center (successor to AWS Single Sign-On)。

- 创建没有权限的用户。
- 创建授予列出账户的 Amazon S3 存储桶的权限的角色
- 添加策略以允许用户代入该角色。
- 代入角色并使用临时凭证列出 S3 存储桶，然后清除资源。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-iam"
require "aws-sdk-s3"

# Wraps the scenario actions.
class ScenarioCreateUserAssumeRole
  attr_reader :iam_resource

  # @param iam_resource [Aws::IAM::Resource] An AWS IAM resource.
  def initialize(iam_resource)
    @iam_resource = iam_resource
  end

  # Waits for the specified number of seconds.
  #
  # @param duration [Integer] The number of seconds to wait.
  def wait(duration)
    puts("Give AWS time to propagate resources...")
    sleep(duration)
  end

  # Creates a user.
  #
  # @param user_name [String] The name to give the user.
  # @return [Aws::IAM::User] The newly created user.
  def create_user(user_name)
    user = @iam_resource.create_user(user_name: user_name)
    puts("Created demo user named #{user.name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Tried and failed to create demo user.")
    puts("\t#{e.code}: #{e.message}")
    puts("\nCan't continue the demo without a user!")
    raise
  else
    user
  end

  # Creates an access key for a user.
  #
  # @param user [Aws::IAM::User] The user that owns the key.
  # @return [Aws::IAM::AccessKeyPair] The newly created access key.
  def create_access_key_pair(user)
    user_key = user.create_access_key_pair
    puts("Created access key pair for user.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't create access keys for user #{user.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    user_key
  end

  # Creates a role that can be assumed by a user.
  #
  # @param role_name [String] The name to give the role.
  # @param user [Aws::IAM::User] The user who is granted permission to assume the role.
  # @return [Aws::IAM::Role] The newly created role.
  def create_role(role_name, user)
    role = @iam_resource.create_role(
```

```
      role_name: role_name,
      assume_role_policy_document: {
        Version: "2012-10-17",
        Statement: [{
          Effect: "Allow",
          Principal: {'AWS': user.arn},
          Action: "sts:AssumeRole"
        }]
      }.to_json)
    puts("Created role #{role.name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't create a role for the demo. Here's why: ")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    role
  end

  # Creates a policy that grants permission to list S3 buckets in the account, and
  # then attaches the policy to a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param role [Aws::IAM::Role] The role that the policy is attached to.
  # @return [Aws::IAM::Policy] The newly created policy.
  def create_and_attach_role_policy(policy_name, role)
    policy = @iam_resource.create_policy(
      policy_name: policy_name,
      policy_document: {
        Version: "2012-10-17",
        Statement: [{
          Effect: "Allow",
          Action: "s3:ListAllMyBuckets",
          Resource: "arn:aws:s3:::*"
        }]
      }.to_json)
    role.attach_policy(policy_arn: policy.arn)
    puts("Created policy #{policy.policy_name} and attached it to role #{role.name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't create a policy and attach it to role #{role.name}. Here's why: ")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    policy
  end

  # Creates an inline policy for a user that lets the user assume a role.
  #
  # @param policy_name [String] The name to give the policy.
  # @param user [Aws::IAM::User] The user that owns the policy.
  # @param role [Aws::IAM::Role] The role that can be assumed.
  # @return [Aws::IAM::UserPolicy] The newly created policy.
  def create_user_policy(policy_name, user, role)
    policy = user.create_policy(
      policy_name: policy_name,
      policy_document: {
        Version: "2012-10-17",
        Statement: [{
          Effect: "Allow",
          Action: "sts:AssumeRole",
          Resource: role.arn
        }]
      }.to_json)
    puts("Created an inline policy for #{user.name} that lets the user assume role
#{role.name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't create an inline policy for user #{user.name}. Here's why: ")
```

```
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    policy
  end

  # Creates an Amazon S3 resource with specified credentials. This is separated into a
  # factory function so that it can be mocked for unit testing.
  #
  # @param credentials [Aws::Credentials] The credentials used by the Amazon S3
resource.
  def create_s3_resource(credentials)
    Aws::S3::Resource.new(client: Aws::S3::Client.new(credentials: credentials))
  end

  # Lists the S3 buckets for the account, using the specified Amazon S3 resource.
  # Because the resource uses credentials with limited access, it may not be able to
  # list the S3 buckets.
  #
  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def list_buckets(s3_resource)
    count = 10
    s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
  rescue Aws::Errors::ServiceError => e
    if e.code == "AccessDenied"
      puts("Attempt to list buckets with no permissions: AccessDenied.")
    else
      puts("Couldn't list buckets for the account. Here's why: ")
      puts("\t#{e.code}: #{e.message}")
      raise
    end
  end

  # Creates an AWS Security Token Service (AWS STS) client with specified credentials.
  # This is separated into a factory function so that it can be mocked for unit
testing.
  #
  # @param key_id [String] The ID of the access key used by the STS client.
  # @param key_secret [String] The secret part of the access key used by the STS
client.
  def create_sts_client(key_id, key_secret)
    Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
  end

  # Gets temporary credentials that can be used to assume a role.
  #
  # @param role_arn [String] The ARN of the role that is assumed when these credentials
  #                          are used.
  # @param sts_client [AWS::STS::Client] An AWS STS client.
  # @return [Aws::AssumeRoleCredentials] The credentials that can be used to assume the
role.
  def assume_role(role_arn, sts_client)
    credentials = Aws::AssumeRoleCredentials.new(
      client: sts_client,
      role_arn: role_arn,
      role_session_name: "create-use-assume-role-scenario"
    )
    puts("Assumed role '#{role_arn}', got temporary credentials.")
    credentials
  end

  # Deletes a role. If the role has policies attached, they are detached and
```

```
  # deleted before the role is deleted.
  #
  # @param role [Aws::IAM::Role] The role to delete.
  def delete_role(role)
    role.attached_policies.each do |policy|
      name = policy.policy_name
      policy.detach_role(role_name: role.name)
      policy.delete
      puts("Deleted policy #{name}.")
    end
    name = role.name
    role.delete
    puts("Deleted role #{name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't detach policies and delete role #{role.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
    raise
  end

  # Deletes a user. If the user has inline policies or access keys, they are deleted
  # before the user is deleted.
  #
  # @param user [Aws::IAM::User] The user to delete.
  def delete_user(user)
    user.policies.each do |policy|
      name = policy.name
      policy.delete
      puts("Deleted user policy #{name}.")
    end
    user.access_keys.each do |key|
      key.delete
      puts("Deleted access key for user #{user.name}.")
    end
    name = user.name
    user.delete
    puts("Deleted user #{name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't detach policies and delete user #{user.name}. Here's why:")
    puts("\t#{e.code}: #{e.message}")
  end
end

# Runs the IAM create a user and assume a role scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the IAM create a user and assume a role demo!")
  puts("-" * 88)

  user = scenario.create_user("doc-example-user-#{Random.uuid}")
  user_key = scenario.create_access_key_pair(user)
  scenario.wait(10)
  role = scenario.create_role("doc-example-role-#{Random.uuid}", user)
  scenario.create_and_attach_role_policy("doc-example-role-policy-#{Random.uuid}",
 role)
  scenario.create_user_policy("doc-example-user-policy-#{Random.uuid}", user, role)
  scenario.wait(10)
  puts("Try to list buckets with credentials for a user who has no permissions.")
  puts("Expect AccessDenied from this call.")
  scenario.list_buckets(
    scenario.create_s3_resource(Aws::Credentials.new(user_key.id, user_key.secret)))
  puts("Now, assume the role that grants permission.")
  temp_credentials = scenario.assume_role(
    role.arn, scenario.create_sts_client(user_key.id, user_key.secret))
  puts("Here are your buckets:")
  scenario.list_buckets(scenario.create_s3_resource(temp_credentials))
  puts("Deleting role '#{role.name}' and attached policies.")
```

```
  scenario.delete_role(role)
  puts("Deleting user '#{user.name}', policies, and keys.")
  scenario.delete_user(user)

  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError => e
  puts("Something went wrong with the demo.")
  puts("\t#{e.code}: #{e.message}")
end

run_scenario(ScenarioCreateUserAssumeRole.new(Aws::IAM::Resource.new)) if $PROGRAM_NAME
 == __FILE__
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey
  - DeletePolicy
  - DeleteRole
  - DeleteUser
  - DeleteUserPolicy
  - DetachRolePolicy
  - PutUserPolicy

使用 Ruby 开发工具包的 Lambda 示例

以下代码示例向您展示了如何使用与 Lambda AWS SDK for Ruby 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)
- 场景 (p. 205)

操作

创建函数

以下代码示例显示了如何创建 Lambda 函数。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Deploys a Lambda function.
  #
  # @param function_name: The name of the Lambda function.
  # @param handler_name: The fully qualified name of the handler function. This
  #                      must include the file name and the function name.
  # @param role_arn: The IAM role to use for the function.
  # @param deployment_package: The deployment package that contains the function
  #                            code in .zip format.
  # @return: The Amazon Resource Name (ARN) of the newly created function.
  def create_function(function_name, handler_name, role_arn, deployment_package)
    response = @lambda_client.create_function({
                                                role: role_arn.to_s,
                                                function_name: function_name,
                                                handler: handler_name,
                                                runtime: "ruby2.7",
                                                code: {
                                                  zip_file: deployment_package
                                                },
                                                environment: {
                                                  variables: {
                                                    "LOG_LEVEL" => "info"
                                                  }
                                                }
                                              })
    @lambda_client.wait_until(:function_active_v2, { function_name: function_name}) do
 |w|
      w.max_attempts = 5
      w.delay = 5
    end
    response
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error creating #{function_name}:\n #{e.message}")
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 CreateFunctionAWS SDK for RubyAPI 参考文档。

删除函数

以下代码示例显示了如何删除 Lambda 函数。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client
```

```
  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Deletes a Lambda function.
  # @param function_name: The name of the function to delete.
  def delete_function(function_name)
    print "Deleting function: #{function_name}..."
    @lambda_client.delete_function(
      function_name: function_name
    )
    print "Done!".green
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error deleting #{function_name}:\n #{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 DeleteFunctionAWS SDK for RubyAPI 参考文档。

获取函数

以下代码示例显示如何获取 Lambda 函数。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Gets data about a Lambda function.
  #
  # @param function_name: The name of the function.
  # @return response: The function data, or nil if no such function exists.
  def get_function(function_name)
    @lambda_client.get_function(
      {
        function_name: function_name
      }
    )
  rescue Aws::Lambda::Errors::ResourceNotFoundException => e
    @logger.debug("Could not find function: #{function_name}:\n #{e.message}")
    nil
  end
```

- 有关 API 的详细信息，请参阅 GetFunctionAWS SDK for RubyAPI 参考文档。

调用函数

以下代码示例显示了如何调用 Lambda 函数。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Invokes a Lambda function.
  # @param function_name [String] The name of the function to invoke.
  # @param payload [nil] Payload containing runtime parameters.
  # @return [Object] The response from the function invocation.
  def invoke_function(function_name, payload = nil)
    params = { function_name: function_name}
    params[:payload] = payload unless payload.nil?
    @lambda_client.invoke(params)
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
```

- 有关 API 详细信息，请参阅 AWS SDK for Ruby API 参考中的 Invoke。

列出函数

以下代码示例显示如何列出 Lambda 函数。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Lists the Lambda functions for the current account.
  def list_functions
    functions = []
    @lambda_client.list_functions.each do |response|
      response["functions"].each do |function|
        functions.append(function["function_name"])
      end
    end
    functions
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 ListFunctionsAWS SDK for RubyAPI 参考文档。

更新函数代码

以下代码示例显示如何更新 Lambda 函数代码。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Updates the code for a Lambda function by submitting a .zip archive that contains
  # the code for the function.

  # @param function_name: The name of the function to update.
  # @param deployment_package: The function code to update, packaged as bytes in
  #                            .zip format.
  # @return: Data about the update, including the status.
  def update_function_code(function_name, deployment_package)
    @lambda_client.update_function_code(
      function_name: function_name,
      zip_file: deployment_package
    )
    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name}) do
 |w|
      w.max_attempts = 5
      w.delay = 5
    end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating function code for: #{function_name}:\n
 #{e.message}")
    nil
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to update:\n #{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 UpdateFunctionCodeAWS SDK for RubyAPI 参考文档。

更新函数配置

以下代码示例显示如何更新 Lambda 函数配置。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
class LambdaWrapper
  attr_accessor :lambda_client

  def initialize
    @lambda_client = Aws::Lambda::Client.new
    @logger = Logger.new($stdout)
    @logger.level = Logger::WARN
  end

  # Updates the environment variables for a Lambda function.
  # @param function_name: The name of the function to update.
  # @param log_level: The log level of the function.
  # @return: Data about the update, including the status.
  def update_function_configuration(function_name, log_level)
    @lambda_client.update_function_configuration({
                                                   function_name: function_name,
                                                   environment: {
                                                     variables: {
                                                       "LOG_LEVEL" => log_level
                                                     }
                                                   }
                                                 })
    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name}) do
 |w|
      w.max_attempts = 5
      w.delay = 5
    end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating configurations for #{function_name}:\n
 #{e.message}")
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end
```

- 有关 API 的详细信息，请参阅 UpdateFunctionConfigurationAWS SDK for RubyAPI 参考文档。

场景

开始使用函数

以下代码示例显示了如何：

- 创建 IAM 角色和 Lambda 函数，然后上传处理程序代码。
- 使用单个参数调用函数并获取结果。
- 更新函数代码并使用环境变量配置。
- 使用新参数调用函数并获取结果。显示返回的执行日志。
- 列出您账户的函数，然后清理函数。

有关更多信息，请参阅使用控制台创建 Lambda 函数。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

为能够写入日志的 Lambda 函数设置必备的 IAM 权限。

```
  # Get an AWS Identity and Access Management (IAM) role.
  #
  # @param iam_role_name: The name of the role to retrieve.
  # @param action: Whether to create or destroy the IAM apparatus.
  # @return: The IAM role.
  def manage_iam(iam_role_name, action)
    role_policy = {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Effect': "Allow",
          'Principal': {
            'Service': "lambda.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }
    case action
    when "create"
      role = $iam_client.create_role(
        role_name: iam_role_name,
        assume_role_policy_document: role_policy.to_json
      )
      $iam_client.attach_role_policy(
        {
          policy_arn: "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole",
          role_name: iam_role_name
        }
      )
      $iam_client.wait_until(:role_exists, { role_name: iam_role_name }) do |w|
        w.max_attempts = 5
        w.delay = 5
      end
      @logger.debug("Successfully created IAM role: #{role['role']['arn']}")
      @logger.debug("Enforcing a 10-second sleep to allow IAM role to activate fully.")
      sleep(10)
      return role, role_policy.to_json
    when "destroy"
      $iam_client.detach_role_policy(
        {
          policy_arn: "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole",
          role_name: iam_role_name
        }
      )
      $iam_client.delete_role(
        role_name: iam_role_name
      )
      @logger.debug("Detached policy & deleted IAM role: #{iam_role_name}")
    else
      raise "Incorrect action provided. Must provide 'create' or 'destroy'"
    end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error creating role or attaching policy:\n
 #{e.message}")
  end
```

定义一个 Lambda 处理程序，用以递增作为调用参数提供的数字。

```
require "logger"
```

```
# A function that increments a whole number by one (1) and logs the result.
# Requires a manually-provided runtime parameter, 'number', which must be Int
#
# @param event [Hash] Parameters sent when the function is invoked
# @param context [Hash] Methods and properties that provide information
# about the invocation, function, and execution environment.
# @return incremented_number [String] The incremented number.
def lambda_handler(event:, context:)
  logger = Logger.new($stdout)
  log_level = ENV["LOG_LEVEL"]
  logger.level = case log_level
                 when "debug"
                   Logger::DEBUG
                 when "info"
                   Logger::INFO
                 else
                   Logger::ERROR
                 end
  logger.debug("This is a debug log message.")
  logger.info("This is an info log message. Code executed successfully!")
  number = event["number"].to_i
  incremented_number = number + 1
  logger.info("You provided #{number.round} and it was incremented to
 #{incremented_number.round}")
  incremented_number.round.to_s
end
```

将 Lambda 函数压缩到部署程序包中。

```
  # Creates a Lambda deployment package in .zip format.
  # This zip can be passed directly as a string to Lambda when creating the function.
  #
  # @param source_file: The name of the object, without suffix, for the Lambda file and
 zip.
  # @return: The deployment package.
  def create_deployment_package(source_file)
    Dir.chdir(File.dirname(__FILE__))
    if File.exist?("lambda_function.zip")
      File.delete("lambda_function.zip")
      @logger.debug("Deleting old zip: lambda_function.zip")
    end
    Zip::File.open("lambda_function.zip", create: true) {
      |zipfile|
      zipfile.add("lambda_function.rb", "#{source_file}.rb")
    }
    @logger.debug("Zipping #{source_file}.rb into: lambda_function.zip.")
    File.read("lambda_function.zip").to_s
  rescue StandardError => e
    @logger.error("There was an error creating deployment package:\n #{e.message}")
  end
```

新建 Lambda 函数。

```
  # Deploys a Lambda function.
  #
  # @param function_name: The name of the Lambda function.
  # @param handler_name: The fully qualified name of the handler function. This
  #                      must include the file name and the function name.
  # @param role_arn: The IAM role to use for the function.
  # @param deployment_package: The deployment package that contains the function
  #                            code in .zip format.
```

```
  # @return: The Amazon Resource Name (ARN) of the newly created function.
  def create_function(function_name, handler_name, role_arn, deployment_package)
    response = @lambda_client.create_function({
                                                role: role_arn.to_s,
                                                function_name: function_name,
                                                handler: handler_name,
                                                runtime: "ruby2.7",
                                                code: {
                                                  zip_file: deployment_package
                                                },
                                                environment: {
                                                  variables: {
                                                    "LOG_LEVEL" => "info"
                                                  }
                                                }
                                              })
    @lambda_client.wait_until(:function_active_v2, { function_name: function_name}) do
|w|
      w.max_attempts = 5
      w.delay = 5
    end
    response
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error creating #{function_name}:\n #{e.message}")
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end
```

使用可选的运行时参数调用 Lambda 函数。

```
  # Invokes a Lambda function.
  # @param function_name [String] The name of the function to invoke.
  # @param payload [nil] Payload containing runtime parameters.
  # @return [Object] The response from the function invocation.
  def invoke_function(function_name, payload = nil)
    params = { function_name: function_name}
    params[:payload] = payload unless payload.nil?
    @lambda_client.invoke(params)
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
```

更新 Lambda 函数的配置，以注入新的环境变量。

```
  # Updates the environment variables for a Lambda function.
  # @param function_name: The name of the function to update.
  # @param log_level: The log level of the function.
  # @return: Data about the update, including the status.
  def update_function_configuration(function_name, log_level)
    @lambda_client.update_function_configuration({
                                                   function_name: function_name,
                                                   environment: {
                                                     variables: {
                                                       "LOG_LEVEL" => log_level
                                                     }
                                                   }
                                                 })
    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name}) do
|w|
      w.max_attempts = 5
      w.delay = 5
```

```
    end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating configurations for #{function_name}:\n
#{e.message}")
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to activate:\n #{e.message}")
  end
```

使用包含不同代码的不同部署程序包更新 Lambda 函数的代码。

```
  # Updates the code for a Lambda function by submitting a .zip archive that contains
  # the code for the function.

  # @param function_name: The name of the function to update.
  # @param deployment_package: The function code to update, packaged as bytes in
  #                            .zip format.
  # @return: Data about the update, including the status.
  def update_function_code(function_name, deployment_package)
    @lambda_client.update_function_code(
      function_name: function_name,
      zip_file: deployment_package
    )
    @lambda_client.wait_until(:function_updated_v2, { function_name: function_name}) do
|w|
      w.max_attempts = 5
      w.delay = 5
    end
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error updating function code for: #{function_name}:\n
#{e.message}")
    nil
  rescue Aws::Waiters::Errors::WaiterFailed => e
    @logger.error("Failed waiting for #{function_name} to update:\n #{e.message}")
  end
```

使用内置分页工具列出所有现有的 Lambda 函数。

```
  # Lists the Lambda functions for the current account.
  def list_functions
    functions = []
    @lambda_client.list_functions.each do |response|
      response["functions"].each do |function|
        functions.append(function["function_name"])
      end
    end
    functions
  rescue Aws::Lambda::Errors::ServiceException => e
    @logger.error("There was an error executing #{function_name}:\n #{e.message}")
  end
```

删除特定的 Lambda 函数。

```
  # Deletes a Lambda function.
  # @param function_name: The name of the function to delete.
  def delete_function(function_name)
    print "Deleting function: #{function_name}..."
    @lambda_client.delete_function(
      function_name: function_name
    )
```

```
  print "Done!".green
rescue Aws::Lambda::Errors::ServiceException => e
  @logger.error("There was an error deleting #{function_name}:\n #{e.message}")
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - CreateFunction
  - DeleteFunction
  - GetFunction
  - 调用
  - ListFunctions
  - UpdateFunctionCode
  - UpdateFunctionConfiguration

使用 Ruby 开发工具包的亚马逊 RDS 示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与 Amazon RDS 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)

操作

创建数据库实例的快照

以下代码示例显示了如何创建 Amazon RDS 数据库实例的快照。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
                                db_snapshot_identifier: id
                              })
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 CreateDBSnapshot。

描述数据库实例

以下代码示例显示了如何描述 Amazon RDS 数据库实例。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
                          "name": i.id,
                          "status": i.db_instance_status
                        })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 DescribeDBInstances。

描述数据库参数组

以下代码示例显示了如何描述 Amazon RDS 数据库参数组。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
```

```
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- 有关 API 的详细信息，请参阅 AWS SDK for RubyAP I 参考中的 describedB ParameterGroups。

描述数据库参数组中的参数

以下代码示例显示如何描述 Amazon RDS 数据库参数组中的参数。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 DescribeDBParameters。

描述数据库实例的快照

以下代码示例显示了如何描述 Amazon RDS 数据库实例的快照。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
```

```
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
                                "id": s.snapshot_id,
                                "status": s.status
                              })
  end
  instance_snapshots
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的 DescribeDBSnapshots。

使用 Ruby 软件开发工具包的亚马逊 S3 示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与 Amazon S3 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)
- 场景 (p. 205)

操作

将 CORS 规则添加到桶

以下代码示例显示如何向 S3 存储桶添加跨源资源共享 (CORS) 规则。

SDK for Ruby

> Note
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
 existing bucket.
  def initialize(bucket_cors)
```

```
    @bucket_cors = bucket_cors
  end

  # Sets CORS rules on a bucket.
  #
  # @param allowed_methods [Array<String>] The types of HTTP requests to allow.
  # @param allowed_origins [Array<String>] The origins to allow.
  # @returns [Boolean] True if the CORS rules were set; otherwise, false.
  def set_cors(allowed_methods, allowed_origins)
    @bucket_cors.put(
      cors_configuration: {
        cors_rules: [
          {
            allowed_methods: allowed_methods,
            allowed_origins: allowed_origins,
            allowed_headers: %w[*],
            max_age_seconds: 3600
          }
        ]
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set CORS rules for #{@bucket_cors.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- 有关 API 的详细信息，请参阅 PutBucketCorsAWS SDK for RubyAPI 参考文档。

将策略添加到桶

以下代码示例显示如何向 S3 存储桶添加策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured with
 an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Sets a policy on a bucket.
  #
  def set_policy(policy)
    @bucket_policy.put(policy: policy)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set the policy for #{@bucket_policy.bucket.name}. Here's why:
 #{e.message}"
    false
```

```
  end

end
```

- 有关 API 的详细信息，请参阅 PutBucketPolicyAWS SDK for RubyAPI 参考文档。

将对象从一个桶复制到另一个

以下代码示例显示如何将 S3 对象从一个存储桶复制到另一个存储桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

复制对象。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is used
 as the source object for
  #                                        copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket and rename it with the target
 key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise, nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Replace the source and target bucket names with existing buckets you own and replace
 the source object key
# with an existing object in the source bucket.
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object
```

```
  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

复制对象并向目标对象添加服务器端加密。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is used
 as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
 key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise, nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
 server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Replace the source and target bucket names with existing buckets you own and replace
 the source object key
# with an existing object in the source bucket.
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key} and "\
       "encrypted the target with #{target_object.server_side_encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 CopyObjectAWS SDK for RubyAPI 参考文档。

创建桶

以下代码示例显示了如何创建 S3 存储桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name. This
 is a client-side object until
  #                                 create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      "None. You must create a bucket before you can get it's location!"
    else
      @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("doc-example-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 CreateBucketAWS SDK for RubyAPI 参考文档。

从桶中删除 CORS 规则

以下代码示例显示如何从 S3 存储桶中删除 CORS 规则。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
 existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Deletes the CORS configuration of a bucket.
  #
  # @return [Boolean] True if the CORS rules were deleted; otherwise, false.
  def delete_cors
    @bucket_cors.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete CORS rules for #{@bucket_cors.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- 有关 API 的详细信息，请参阅 DeleteBucketCorsAWS SDK for RubyAPI 参考文档。

从桶中删除策略

以下代码示例显示如何从 S3 存储桶中删除策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured with
 an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  def delete_policy
```

```
    @bucket_policy.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete the policy from #{@bucket_policy.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- 有关 API 的详细信息，请参阅 DeleteBucketPolicyAWS SDK for RubyAPI 参考文档。

删除空桶

以下代码示例显示如何删除空的 S3 存储桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
  answer = gets.chomp.downcase
  if answer == "y"
    bucket.objects.batch_delete!
    bucket.delete
    puts("Emptied and deleted bucket #{bucket.name}.\n")
  end
rescue Aws::Errors::ServiceError => e
  puts("Couldn't empty and delete bucket #{bucket.name}.")
  puts("\t#{e.code}: #{e.message}")
  raise
end
```

- 有关 API 的详细信息，请参阅 DeleteBucketAWS SDK for RubyAPI 参考文档。

删除多个对象

以下代码示例显示如何从 S3 存储桶中删除多个对象。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Deletes the objects in an Amazon S3 bucket and deletes the bucket.
#
# @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
def delete_bucket(bucket)
  puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
  answer = gets.chomp.downcase
```

```
    if answer == "y"
      bucket.objects.batch_delete!
      bucket.delete
      puts("Emptied and deleted bucket #{bucket.name}.\n")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't empty and delete bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
```

- 有关 API 的详细信息，请参阅 DeleteObjectsAWS SDK for RubyAPI 参考文档。

确定对象是否存在及其内容类型

以下代码示例显示如何确定 S3 存储桶中对象的存在和内容类型。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectExistsWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Checks whether the object exists.
  #
  # @return [Boolean] True if the object exists; otherwise false.
  def exists?
    @object.exists?
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't check existence of object #{@object.bucket.name}:#{@object.key}.
 Here's why: #{e.message}"
    false
  end
end

# Replace bucket name and object key with an existing bucket and object that you own.
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectExistsWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  exists = wrapper.exists?

  puts "Object #{object_key} #{exists ? 'does' : 'does not'} exist."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 HeadObjectAWS SDK for RubyAPI 参考文档。

获取桶的 CORS 规则

以下代码示例显示如何获取 S3 存储桶的跨源资源共享 (CORS) 规则。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with an
 existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Gets the CORS configuration of a bucket.
  #
  # @return [Aws::S3::Type::GetBucketCorsOutput, nil] The current CORS configuration
 for the bucket.
  def get_cors
    @bucket_cors.data
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's why:
 #{e.message}"
    nil
  end

end
```

- 有关 API 的详细信息，请参阅 GetBucketCorsAWS SDK for RubyAPI 参考文档。

从桶中获取对象

以下代码示例显示如何从 S3 存储桶中的对象读取数据。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

获取对象。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end
```

```
  # Gets the object directly to a file.
  #
  # @param target_path [String] The path to the file where the object is downloaded.
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def get_object(target_path)
    @object.get(response_target: target_path)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Replace bucket name and object key with an existing bucket and object that you own.
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"
  target_path = "my-object-as-file.txt"

  wrapper = ObjectGetWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  obj_data = wrapper.get_object(target_path)
  return unless obj_data

  puts "Object #{object_key} (#{obj_data.content_length} bytes} downloaded to
 #{target_path}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

获取对象并报告其服务器端加密状态。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def get_object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Replace bucket name and object key with an existing bucket and object that you own.
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
 obj_data.server_side_encryption
```

```
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 GetObjectAWS SDK for RubyAPI 参考文档。

获取桶的策略

以下代码示例显示如何获取 S3 存储桶的策略。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object configured with
 an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Gets the policy of a bucket.
  #
  # @return [Aws::S3::GetBucketPolicyOutput, nil] The current bucket policy.
  def get_policy
    policy = @bucket_policy.data.policy
    policy.respond_to?(:read) ? policy.read : policy
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get the policy for #{@bucket_policy.bucket.name}. Here's why:
 #{e.message}"
    nil
  end

end
```

- 有关 API 的详细信息，请参阅 GetBucketPolicyAWS SDK for RubyAPI 参考文档。

列出桶

以下代码示例显示了如何列出 S3 存储桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
```

```
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
    false
  end
end

def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 ListBucketsAWS SDK for RubyAPI 参考文档。

列出桶中的对象

以下代码示例显示如何列出 S3 存储桶中的对象。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketListObjectsWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
    count = 0
    puts "The objects in #{@bucket.name} are:"
```

```
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{bucket.name}. Here's why: #{e.message}"
    0
  end
end

def run_demo
  bucket_name = "doc-example-bucket"

  wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
  count = wrapper.list_objects(25)
  puts "Listed #{count} objects."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 ListObjectsAWS SDK for RubyAPI 参考文档。

设置桶的网站配置

以下代码示例显示如何为 S3 存储桶设置网站配置。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket website actions.
class BucketWebsiteWrapper
  attr_reader :bucket_website

  # @param bucket_website [Aws::S3::BucketWebsite] A bucket website object configured
 with an existing bucket.
  def initialize(bucket_website)
    @bucket_website = bucket_website
  end

  # Sets a bucket as a static website.
  #
  # @param index_document [String] The name of the index document for the website.
  # @param error_document [String] The name of the error document to show for 4XX
 errors.
  # @return [Boolean] True when the bucket is configured as a website; otherwise,
 false.
  def set_website(index_document, error_document)
    @bucket_website.put(
      website_configuration: {
        index_document: { suffix: index_document },
        error_document: { key: error_document }
      }
    )
    true
```

```
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's why:
 #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  index_document = "index.html"
  error_document = "404.html"

  wrapper = BucketWebsiteWrapper.new(Aws::S3::BucketWebsite.new(bucket_name))
  return unless wrapper.set_website(index_document, error_document)

  puts "Successfully configured bucket #{bucket_name} as a static website."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 PutBucketWebsiteAWS SDK for RubyAPI 参考文档。

将对象上传到桶

以下代码示例显示如何将对象上传到 S3 存储桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

使用托管上传工具 (Object.upload_file) 上传文件。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
 #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-uploaded-file"
```

```
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

使用 Object.put 上载文件。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

使用 Object.put 上载文件并添加服务器端加密。

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end
```

```
  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name, object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 PutObjectAWS SDK for RubyAPI 参考文档。

场景

创建预签名 URL

以下代码示例显示如何为 Amazon S3 创建预签名 URL 并上传对象。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
def get_presigned_url(bucket, object_key)
  url = bucket.object(object_key).presigned_url(:put)
  puts "Created presigned URL: #{url}"
  URI(url)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's why:
 #{e.message}"
end

def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-file.txt"
  object_content = "This is the content of my-file.txt."

  bucket = Aws::S3::Bucket.new(bucket_name)
```

```
  presigned_url = get_presigned_url(bucket, object_key)
  return unless presigned_url

  response = Net::HTTP.start(presigned_url.host) do |http|
    http.send_request("PUT", presigned_url.request_uri, object_content, "content_type"
 => "")
  end

  case response
  when Net::HTTPSuccess
    puts "Content uploaded!"
  else
    puts response.value
  end
end

run_demo if $PROGRAM_NAME == __FILE__
```

开始使用桶和对象

以下代码示例显示了如何：

- 创建桶并将文件上传到其中。
- 从桶下载对象。
- 将对象复制到桶中的子文件夹。
- 列出桶中的对象。
- 删除桶对象和桶。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-s3"

# Wraps the getting started scenario actions.
class ScenarioGettingStarted
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Creates a bucket with a random name in the currently configured account and
  # AWS Region.
  #
  # @return [Aws::S3::Bucket] The newly created bucket.
  def create_bucket
    bucket = @s3_resource.create_bucket(
      bucket: "doc-example-bucket-#{Random.uuid}",
      create_bucket_configuration: {
        location_constraint: "us-east-2" # Note: only certain regions permitted
      }
    )
    puts("Created demo bucket named #{bucket.name}.")
  rescue Aws::Errors::ServiceError => e
```

```
    puts("Tried and failed to create demo bucket.")
    puts("\t#{e.code}: #{e.message}")
    puts("\nCan't continue the demo without a bucket!")
    raise
  else
    bucket
  end

  # Requests a file name from the user.
  #
  # @return The name of the file.
  def create_file
    File.open("demo.txt", w) { |f| f.write("This is a demo file.") }
  end

  # Uploads a file to an Amazon S3 bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket object representing the upload
 destination
  # @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded file.
  def upload_file(bucket)
    File.open("demo.txt", "w+") { |f| f.write("This is a demo file.") }
    s3_object = bucket.object(File.basename("demo.txt"))
    s3_object.upload_file("demo.txt")
    puts("Uploaded file demo.txt into bucket #{bucket.name} with key
 #{s3_object.key}.")
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't upload file demo.txt to #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    s3_object
  end

  # Downloads an Amazon S3 object to a file.
  #
  # @param s3_object [Aws::S3::Object] The object to download.
  def download_file(s3_object)
    puts("\nDo you want to download #{s3_object.key} to a local file (y/n)? ")
    answer = gets.chomp.downcase
    if answer == "y"
      puts("Enter a name for the downloaded file: ")
      file_name = gets.chomp
      s3_object.download_file(file_name)
      puts("Object #{s3_object.key} successfully downloaded to #{file_name}.")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't download #{s3_object.key}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end

  # Copies an Amazon S3 object to a subfolder within the same bucket.
  #
  # @param source_object [Aws::S3::Object] The source object to copy.
  # @return [Aws::S3::Object, nil] The destination object.
  def copy_object(source_object)
    dest_object = nil
    puts("\nDo you want to copy #{source_object.key} to a subfolder in your bucket (y/
n)? ")
    answer = gets.chomp.downcase
    if answer == "y"
      dest_object = source_object.bucket.object("demo-folder/#{source_object.key}")
      dest_object.copy_from(source_object)
      puts("Copied #{source_object.key} to #{dest_object.key}.")
    end
```

```
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't copy #{source_object.key}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    dest_object
  end

  # Lists the objects in an Amazon S3 bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket to query.
  def list_objects(bucket)
    puts("\nYour bucket contains the following objects:")
    bucket.objects.each do |obj|
      puts("\t#{obj.key}")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't list the objects in bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end

  # Deletes the objects in an Amazon S3 bucket and deletes the bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
  def delete_bucket(bucket)
    puts("\nDo you want to delete all of the objects as well as the bucket (y/n)? ")
    answer = gets.chomp.downcase
    if answer == "y"
      bucket.objects.batch_delete!
      bucket.delete
      puts("Emptied and deleted bucket #{bucket.name}.\n")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't empty and delete bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
end

# Runs the Amazon S3 getting started scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the Amazon S3 getting started demo!")
  puts("-" * 88)

  bucket = scenario.create_bucket
  s3_object = scenario.upload_file(bucket)
  scenario.download_file(s3_object)
  scenario.copy_object(s3_object)
  scenario.list_objects(bucket)
  scenario.delete_bucket(bucket)

  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo!")
end

run_scenario(ScenarioGettingStarted.new(Aws::S3::Resource.new)) if $PROGRAM_NAME ==
 __FILE__
```

- 有关 API 详细信息，请参阅《AWS SDK for Ruby API 参考》中的以下主题。
  - CopyObject

- CreateBucket
- DeleteBucket
- DeleteObjects
- GetObject
- ListObjects
- PutObject

使用 Ruby 开发工具包的亚马逊 SNS 示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与 Amazon SNS 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)

操作

创建主题

以下代码示例显示了如何创建 Amazon SNS 主题。

SDK for Ruby

> Note
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

def topic_created?(sns_client, topic_name)

  sns_client.create_topic(name: topic_name)
  rescue StandardError => e
    puts "Error while creating the topic named '#{topic_name}': #{e.message}"
end

# Full example call:
def run_me
  topic_name = "TOPIC_NAME"
  region = "REGION"

  sns_client = Aws::SNS::Client.new(region: region)

  puts "Creating the topic '#{topic_name}'..."

  if topic_created?(sns_client, topic_name)
    puts "The topic was created."
  else
    puts "The topic was not created. Stopping program."
```

```
    exit 1
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 的详细信息，请参阅 CreateTopicAWS SDK for RubyAPI 参考文档。

列出主题的订阅者

以下代码示例显示如何检索 Amazon SNS 主题的订阅者列表。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

def show_subscriptions?(sns_client, topic_arn)
  topic = sns_client.topic(topic_arn)
  topic.subscriptions.each do |s|
    puts s.attributes["Endpoint"]
  end

rescue StandardError => e
  puts "Error while sending the message: #{e.message}"
end

def run_me

  topic_arn = "SNS_TOPIC_ARN"
  region = "REGION"

  sns_client = Aws::SNS::Resource.new(region: region)

  puts "Listing subscriptions to the topic."

  if show_subscriptions?(sns_client, topic_arn)
  else
    puts "There was an error. Stopping program."
    exit 1
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 的详细信息，请参阅 ListSubscriptionsAWS SDK for RubyAPI 参考文档。

列出主题

以下代码示例显示了如何列出 Amazon SNS 主题。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

def list_topics?(sns_client)
  sns_client.topics.each do |topic|
    puts topic.arn
rescue StandardError => e
  puts "Error while listing the topics: #{e.message}"
  end
end

def run_me

  region = "REGION"
  sns_client = Aws::SNS::Resource.new(region: region)

  puts "Listing the topics."

  if list_topics?(sns_client)
  else
    puts "The bucket was not created. Stopping program."
    exit 1
  end
end
run_me if $PROGRAM_NAME == __FILE__
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 的详细信息，请参阅 ListTopicsAWS SDK for RubyAPI 参考文档。

向主题发布

以下代码示例显示了如何向 Amazon SNS 主题发布消息。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

def message_sent?(sns_client, topic_arn, message)

  sns_client.publish(topic_arn: topic_arn, message: message)
rescue StandardError => e
  puts "Error while sending the message: #{e.message}"
end

def run_me

  topic_arn = "SNS_TOPIC_ARN"
```

```
  region = "REGION"
  message = "MESSAGE" # The text of the message to send.

  sns_client = Aws::SNS::Client.new(region: region)

  puts "Message sending."

  if message_sent?(sns_client, topic_arn, message)
    puts "The message was sent."
  else
    puts "The message was not sent. Stopping program."
    exit 1
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 详细信息，请参阅 AWS SDK for Ruby API 参考中的 Publish。

设置主题属性

以下代码示例显示了如何设置 Amazon SNS 主题属性。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

policy  = '{
  "Version":"2008-10-17",
  "Id":"__default_policy_ID",
  "Statement":[{
    "Sid":"__default_statement_ID",
    "Effect":"Allow",
    "Principal":{
      "AWS":"*"
    },
    "Action":["SNS:Publish"],
    "Resource":"' + MY_TOPIC_ARN + '",
    "Condition":{
      "ArnEquals":{
        "AWS:SourceArn":"' + MY_RESOURCE_ARN + '"}
     }
  }]
}'
# Replace us-west-2 with the AWS Region you're using for Amazon SNS.
sns = Aws::SNS::Resource.new(region: "REGION")

# Get topic by ARN
topic = sns.topic()

# Add policy to topic
topic.set_attributes({
  attribute_name: "POLICY_NAME",
  attribute_value: policy
```

```
})
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 的详细信息，请参阅 SetTopicAttributesAWS SDK for RubyAPI 参考文档。

订阅主题的电子邮件地址

以下代码示例显示如何订阅 Amazon SNS 主题的电子邮件地址。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sns"  # v2: require 'aws-sdk'

def subscription_created?(sns_client, topic_arn, protocol, endpoint)

  sns_client.subscribe(topic_arn: topic_arn, protocol: protocol, endpoint: endpoint)

rescue StandardError => e
  puts "Error while creating the subscription: #{e.message}"
end

# Full example call:
def run_me

  protocol = "email"
endpoint = "EMAIL_ADDRESS"
topic_arn = "TOPIC_ARN"
region = "REGION"

sns_client = Aws::SNS::Client.new(region: region)

puts "Creating the subscription."

  if subscription_created?(sns_client, topic_arn, protocol, endpoint)
    puts "The subscriptions was created."
  else
    puts "The subscription was not created. Stopping program."
    exit 1
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关更多信息，请参阅 AWS SDK for Ruby 开发人员指南。
- 有关 API 详细信息，请参阅 AWS SDK for Ruby API 参考中的 Subscribe。

使用适用于 Ruby 开发工具包的亚马逊 SQS 示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与 Amazon SQS 一起执行操作和实现常见场景。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)

操作

更改消息超时可见性

以下代码示例显示了如何更改 Amazon SQS 消息超时可见性。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"  # v2: require 'aws-sdk'
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
sqs = Aws::SQS::Client.new(region: "us-west-2")

begin
  queue_name = "my-queue"
  queue_url = sqs.get_queue_url(queue_name: queue_name).queue_url

  receive_message_result_before = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10 # Receive up to 10 messages, if there are that many.
  })

  puts "Before attempting to change message visibility timeout: received
 #{receive_message_result_before.messages.count} message(s)."

  receive_message_result_before.messages.each do |message|
    sqs.change_message_visibility({
      queue_url: queue_url,
      receipt_handle: message.receipt_handle,
      visibility_timeout: 30 # This message will not be visible for 30 seconds after
 first receipt.
    })
  end

  # Try to retrieve the original messages after setting their visibility timeout.
  receive_message_result_after = sqs.receive_message({
    queue_url: queue_url,
    max_number_of_messages: 10
  })

  puts "\nAfter attempting to change message visibility timeout: received
 #{receive_message_result_after.messages.count} message(s)."

rescue Aws::SQS::Errors::NonExistentQueue
  puts "Cannot receive messages for a queue named '#{receive_queue_name}', as it does
 not exist."
```

```
end
```

- 有关 API 的详细信息，请参阅 ChangeMessageVisibilityAWS SDK for RubyAPI 参考文档。

创建队列

以下代码示例显示了如何创建 Amazon SQS 队列。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
# This code example demonstrates how to create a queue in Amazon Simple Queue Service
 (Amazon SQS).

require "aws-sdk-sqs"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_name [String] The name of the queue.
# @return [Boolean] true if the queue was created; otherwise, false.
# @example
#   exit 1 unless queue_created?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'my-queue'
#   )
def queue_created?(sqs_client, queue_name)
  sqs_client.create_queue(queue_name: queue_name)
  true
rescue StandardError => e
  puts "Error creating queue: #{e.message}"
  false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Creating the queue named '#{queue_name}'..."

  if queue_created?(sqs_client, queue_name)
    puts "Queue created."
  else
    puts "Queue not created."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 CreateQueueAWS SDK for RubyAPI 参考文档。

删除队列

以下代码示例显示了如何删除 Amazon SQS 队列。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"  # v2: require 'aws-sdk'
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
sqs = Aws::SQS::Client.new(region: "us-west-2")

sqs.delete_queue(queue_url: URL)
```

- 有关 API 的详细信息，请参阅 DeleteQueueAWS SDK for RubyAPI 参考文档。

列出队列

以下代码示例显示了如何列出 Amazon SQS 队列。

SDK for Ruby

Note

还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @example
#   list_queue_urls(Aws::SQS::Client.new(region: 'us-west-2'))
def list_queue_urls(sqs_client)
  queues = sqs_client.list_queues

  queues.queue_urls.each do |url|
    puts url
  end
rescue StandardError => e
  puts "Error listing queue URLs: #{e.message}"
end

# Lists the attributes of a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @example
#   list_queue_attributes(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
#   )
def list_queue_attributes(sqs_client, queue_url)
  attributes = sqs_client.get_queue_attributes(
    queue_url: queue_url,
    attribute_names: ["All"]
  )

  attributes.attributes.each do |key, value|
    puts "#{key}: #{value}"
  end
```

```
rescue StandardError => e
  puts "Error getting queue attributes: #{e.message}"
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Listing available queue URLs..."
  list_queue_urls(sqs_client)

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  puts "\nGetting information about queue '#{queue_name}'..."
  list_queue_attributes(sqs_client, queue_url)
end
```

- 有关 API 的详细信息，请参阅 ListQueuesAWS SDK for RubyAPI 参考文档。

从队列接收消息

以下代码示例显示了如何从 Amazon SQS 队列接收消息。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# Receives messages in a queue in Amazon Simple Queue Service (Amazon SQS).
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param max_number_of_messages [Integer] The maximum number of messages
#   to receive. This number must be 10 or less. The default is 10.
# @example
#   receive_messages(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     10
#   )
def receive_messages(sqs_client, queue_url, max_number_of_messages = 10)

  if max_number_of_messages > 10
    puts "Maximum number of messages to receive must be 10 or less. " \
      "Stopping program."
    return
```

```
  end

  response = sqs_client.receive_message(
    queue_url: queue_url,
    max_number_of_messages: max_number_of_messages
  )

  if response.messages.count.zero?
    puts "No messages to receive, or all messages have already " \
      "been previously received."
    return
  end

  response.messages.each do |message|
    puts "-" * 20
    puts "Message body: #{message.body}"
    puts "Message ID:   #{message.message_id}"
  end

rescue StandardError => e
  puts "Error receiving messages: #{e.message}"
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  max_number_of_messages = 10

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Receiving messages from queue '#{queue_name}'..."

  receive_messages(sqs_client, queue_url, max_number_of_messages)
end

run_me if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 ReceiveMessageAWS SDK for RubyAPI 参考文档。

向队列发送一批消息

以下代码示例显示如何向 Amazon SQS 队列发送一批消息。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"
```

```
#
# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param entries [Hash] The contents of the messages to be sent,
#   in the correct format.
# @return [Boolean] true if the messages were sent; otherwise, false.
# @example
#   exit 1 unless messages_sent?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     [
#       {
#         id: 'Message1',
#         message_body: 'This is the first message.'
#       },
#       {
#         id: 'Message2',
#         message_body: 'This is the second message.'
#       }
#     ]
#   )
def messages_sent?(sqs_client, queue_url, entries)
  sqs_client.send_message_batch(
    queue_url: queue_url,
    entries: entries
  )
  true
rescue StandardError => e
  puts "Error sending messages: #{e.message}"
  false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  entries = [
    {
      id: "Message1",
      message_body: "This is the first message."
    },
    {
      id: "Message2",
      message_body: "This is the second message."
    }
  ]

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending messages to the queue named '#{queue_name}'..."

  if messages_sent?(sqs_client, queue_url, entries)
    puts "Messages sent."
  else
    puts "Messages not sent."
  end
end
```

- 有关 API 的详细信息，请参阅 SendMessageBatchAWS SDK for RubyAPI 参考文档。

向队列发送消息

以下代码示例显示了如何向 Amazon SQS 队列发送消息。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
require "aws-sdk-sqs"
require "aws-sdk-sts"

# @param sqs_client [Aws::SQS::Client] An initialized Amazon SQS client.
# @param queue_url [String] The URL of the queue.
# @param message_body [String] The contents of the message to be sent.
# @return [Boolean] true if the message was sent; otherwise, false.
# @example
#   exit 1 unless message_sent?(
#     Aws::SQS::Client.new(region: 'us-west-2'),
#     'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue',
#     'This is my message.'
#   )
def message_sent?(sqs_client, queue_url, message_body)
  sqs_client.send_message(
    queue_url: queue_url,
    message_body: message_body
  )
  true
rescue StandardError => e
  puts "Error sending message: #{e.message}"
  false
end

# Full example call:
# Replace us-west-2 with the AWS Region you're using for Amazon SQS.
def run_me
  region = "us-west-2"
  queue_name = "my-queue"
  message_body = "This is my message."

  sts_client = Aws::STS::Client.new(region: region)

  # For example:
  # 'https://sqs.us-west-2.amazonaws.com/111111111111/my-queue'
  queue_url = "https://sqs." + region + ".amazonaws.com/" +
    sts_client.get_caller_identity.account + "/" + queue_name

  sqs_client = Aws::SQS::Client.new(region: region)

  puts "Sending a message to the queue named '#{queue_name}'..."

  if message_sent?(sqs_client, queue_url, message_body)
    puts "Message sent."
  else
    puts "Message not sent."
  end
end
```

```
run_me if $PROGRAM_NAME == __FILE__
```

- 有关 API 的详细信息，请参阅 SendMessageAWS SDK for RubyAPI 参考文档。

AWS STS使用 Ruby 软件开发工具包的示例

以下代码示例向您展示了如何使用AWS SDK for Ruby与执行操作和实现常见场景AWS STS。

操作是大型程序的代码摘录，必须在上下文中运行。虽然操作向您展示了如何调用单个服务函数，但您可以在其相关场景和跨服务示例中查看操作的上下文。

场景是展示如何通过在同一服务中调用多个函数来完成特定任务的代码示例。

每个示例都包含一个链接GitHub，您可以在其中找到有关如何在上下文中设置和运行代码的说明。

主题

- 操作 (p. 194)

操作

代入角色

以下代码示例显示了如何代入角色AWS STS。

SDK for Ruby

> **Note**
>
> 还有更多GitHub。查找完整示例，学习如何在 AWS 代码示例存储库中进行设置和运行。

```
 # Creates an AWS Security Token Service (AWS STS) client with specified credentials.
 # This is separated into a factory function so that it can be mocked for unit
testing.
 #
 # @param key_id [String] The ID of the access key used by the STS client.
 # @param key_secret [String] The secret part of the access key used by the STS
client.
 def create_sts_client(key_id, key_secret)
   Aws::STS::Client.new(access_key_id: key_id, secret_access_key: key_secret)
 end

 # Gets temporary credentials that can be used to assume a role.
 #
 # @param role_arn [String] The ARN of the role that is assumed when these credentials
 #                          are used.
 # @param sts_client [AWS::STS::Client] An AWS STS client.
 # @return [Aws::AssumeRoleCredentials] The credentials that can be used to assume the
role.
 def assume_role(role_arn, sts_client)
   credentials = Aws::AssumeRoleCredentials.new(
     client: sts_client,
     role_arn: role_arn,
     role_session_name: "create-use-assume-role-scenario"
   )
   puts("Assumed role '#{role_arn}', got temporary credentials.")
   credentials
```

```
end
```

- 有关 API 的详细信息，请参阅 AssumeRoleAWS SDK for RubyAPI 参考文档。

适用于 Ruby AWS 的 SDK 的安全

云安全性一直是 Amazon Web Services (AWS) 的重中之重。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。安全性是AWS和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性。

云安全 — AWS 负责保护运行AWS云中提供的所有服务的基础架构，并为您提供可以安全使用的服务。安全责任是AWS的最高优先任务，作为 AWS合规性计划的一部分，我们安全性的有效性由第三方审计员定期进行测试和验证。

云端安全 — 您的责任由AWS 服务您正在使用的内容以及其他因素决定，包括数据的敏感性、贵组织的要求以及适用的法律和法规。

主题

- 适用于 Ruby AWS 的 SDK 中的数据保护 (p. 307)
- 适用于 Ruby 的 AWS SDK 的身份和访问管理 (p. 308)
- 适用于 Ruby 的 AWS SDK 的合规性 (p. 308)
- 适用于 Ruby AWS 的 SDK 的弹 (p. 308)
- 适用于 Ruby 的 AWS SDK 的基础设施安全 (p. 309)
- 在 Ruby AWS 软件开发工具包中强制使用最低 TLS 版本 (p. 309)
- 亚马逊 S3 加密客户端迁移 (p. 310)

适用于 Ruby AWS 的 SDK 中的数据保护

分AWS担责任模型适用于中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS Cloud 的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。此内容包括您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center (successor to AWS Single Sign-On) 或 AWS Identity and Access Management (IAM) 设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护您的数据：

- 对每个账户使用 Multi-Factor Authentication (MFA)。
- 使用 SSL/TLS 与 AWS 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用 AWS CloudTrail 设置 API 和用户活动日志记录。
- 使用 AWS 加密解决方案以及 AWS 服务 中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如 Name（名称）字段）。这包括当您AWS 服务使用控制台、API 或 AWS SDK 或以其他方式使用时。AWS CLI您在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，我们强烈建议您不要在 URL 中包含凭证信息来验证您对该服务器的请求。

适用于 Ruby 的 AWS SDK 的身份和访问管理

AWS Identity and Access Management (IAM) 是一项 Amazon Web Services (AWS) 服务，可以帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）来使用 AWS 服务 资源。IAM 是一项无需额外费用即可使用的 AWS 服务。

要使用AWS适用于 Ruby 的 SDK 进行访问AWS，你需要一个AWS账户和AWS证书。为了提高 AWS 账户的安全性，我们建议您使用 IAM 用户 来提供访问凭证，而不使用 AWS 账户凭证。

有关使用 IAM 的详细信息，请参阅 IAM。

有关 IAM 用户以及它们对于账户的安全性为何十分重要的概览，请参阅 Amazon Web Services 一般参考中的 AWS 安全凭证。

AWS适用于 Ruby 的 SDK 通过其支持的特定亚马逊网络服务 (AWS) 服务遵循分担责任模式。有关 AWS 服务安全性信息，请参阅 AWS 服务 安全性文档页面和合规性计划规定的 AWS 合规性工作范围内的 AWS 服务。

适用于 Ruby 的 AWS SDK 的合规性

AWS适用于 Ruby 的 SDK 通过其支持的特定亚马逊网络服务 (AWS) 服务遵循分担责任模式。有关 AWS 服务安全性信息，请参阅 AWS 服务 安全性文档页面和合规性计划规定的 AWS 合规性工作范围内的 AWS 服务。

作为多个AWS合规性计划的一部分，第三方审计员将评估亚马逊云科技（AWS）服务的安全性和合规性。其中包括 SOC、PCI、FedRAMP、HIPAA 等。 AWS按合规计划提供了一份AWS 服务经常更新的特定合规计划范围内的AWS服务列表。

提供第三方审计报告，可使用 AWS Artifact 下载。有关更多信息，请参阅下载 AWS Artifact 中的报告。

有关 AWS 合规性计划的更多信息，请参阅 AWS 合规性计划。

使用 AWS SDK for Ruby 访问时，您的合规责任AWS 服务由数据的敏感性、您所在组织的合规目标以及适用的法律和法规决定。如果您对的使用必须符合 HIPAA、PCI 或 FedRAMP 等标准，请AWS提供资源来帮助您：AWS 服务

- 安全与合规性快速入门指南 — 部署指南，讨论架构注意事项，并提供在上部署以安全为重点和以合规为重点的基准环境的步骤。AWS
- HIPAA 安全与合规性架构白皮书 — 介绍公司如何创建符合 HIPA AWS A 的应用程序的白皮书。
- AWS合规资源 — 可能适用于您的行业和地点的工作簿和指南的集合。
- AWS配置-一项评估您的资源配置在多大程度上符合内部惯例、行业准则和法规的服务。
- AWSSecurity Hub — 全面了解您的安全状态AWS，可帮助您检查是否符合安全行业标准和最佳实践。

适用于 Ruby AWS 的 SDK 的弹

Amazon Web Services (AWS) 全球基础设施是围绕AWS 区域可用区构建的。

AWS 区域 提供多个在物理上独立且隔离的可用区，这些可用区与延迟率低、吞吐量高且冗余性高的网络连接在一起。

利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅 AWS 全球基础设施。

AWS适用于 Ruby 的 SDK 通过其支持的特定亚马逊网络服务 (AWS) 服务遵循分担责任模式。有关 AWS 服务安全性信息，请参阅 AWS 服务 安全性文档页面和合规性计划规定的 AWS 合规性工作范围内的 AWS 服务。

适用于 Ruby 的 AWS SDK 的基础设施安全

AWS适用于 Ruby 的 SDK 通过其支持的特定亚马逊网络服务 (AWS) 服务遵循分担责任模式。有关 AWS 服务安全性信息，请参阅 AWS 服务 安全性文档页面和合规性计划规定的 AWS 合规性工作范围内的 AWS 服务。

有关AWS安全流程的信息，请参阅 AWS：安全流程概述白皮书。

在 Ruby AWS 软件开发工具包中强制使用最低 TLS 版本

AWS适用于 Ruby AWS 的 SDK 与使用安全套接字层 (SSL) 或传输层安全 (TLS) 进行保护。所有版本的 SSL 以及低于 1.2 的 TLS 版本都存在可能危及您与之通信安全的漏洞AWS。出于这个原因，你应该确保使用适用于 Ruby 的 AWS SDK 和支持 TLS 版本 1.2 或更高版本的 Ruby。

Ruby 使用 OpenSSL 库来保护 HTTP 连接。通过系统包管理器（、等）、官方安装程序或 Ruby 管理器（rbenv yumapt、RVM 等）安装的支持的 Ruby 版本（1.9.3 及更高版本）通常包含支持 TLS 1.2 的 OpenSSL 1.0.1 或更高版本。

当与 OpenSSL 1.0.1 或更高版本的支持版本的 Ruby 一起使用时，AWS适用于 Ruby 的 SDK 首选 TLS 1.2，并使用客户端和服务器都支持的最新版本的 SSL 或 TLS。对于 TLS 1.2，这始终是至少 TLS 1.2 AWS 服务。（软件开发工具包使用的 Ruby `Net::HTTP` 类与`use_ssl=true`。）

检查 OpenSSL 版本

要确保你安装的 Ruby 使用的是 OpenSSL 1.0.1 或更高版本，请输入以下命令。

```
ruby -r openssl -e 'puts OpenSSL::OPENSSL_VERSION'
```

获取 OpenSSL 版本的另一种方法是直接查询`openssl`可执行文件。首先，使用以下命令找到相应的可执行文件。

```
ruby -r rbconfig -e 'puts RbConfig::CONFIG["configure_args"]'
```

输出应该`--with-openssl-dir=/path/to/openssl`指示 OpenSSL 的安装位置。记下这条路径。要检查 OpenSSL 的版本，请输入以下命令。

```
cd /path/to/openssl
bin/openssl version
```

后一种方法可能不适用于所有 Ruby 安装。

升级 TLS 支持

如果您的 Ruby 安装使用的 OpenSSL 版本低于 1.0.1，请按照 Ruby 安装指南中的说明，使用系统包管理器、Ruby 安装程序或 Ruby 管理器升级 Ruby 或 OpenSSL 安装。如果你要从源代码安装 Ruby，请

先安装最新的 OpenSSL，然后`--with-openssl-dir=/path/to/upgraded/openssl`在运行`./configure`时通过。

亚马逊 S3 加密客户端迁移

本主题介绍如何将您的应用程序从亚马逊简单存储服务 (Amazon S3) 加密客户端的版本 1 (V1) 迁移到版本 2 (V2)，并在整个迁移过程中确保应用程序的可用性。

迁移概述

这种迁移分两个阶段进行：

1. 更新现有客户端以读取新格式。首先，将适用于 Ruby 的 AWS SDK 的更新版本部署到您的应用程序。这将允许现有的 V1 加密客户端解密由新 V2 客户端写入的对象。如果您的应用程序使用多个 AWS SDK，则必须单独升级每个 SDK。

2. 将加密和解密客户端迁移到 V2。一旦所有 V1 加密客户端都能读取新格式，就可以将现有的加密和解密客户端迁移到各自的 V2 版本。

更新现有客户端以读取新格式

V2 加密客户端使用旧版本的客户端不支持的加密算法。迁移的第一步是将您的 V1 解密客户端更新到最新的 SDK 版本。完成此步骤后，您的应用程序的 V1 客户端将能够解密由 V2 加密客户端加密的对象。有关适用于 Ruby 的 AWS SDK 的每个主要版本的详细信息，请参阅下文。

更新 Ruby 版本 3 的 AWS SDK

版本 3 是适用于 Ruby 的 AWS SDK 的最新版本。要完成此迁移，您需要使用 Gem 的 1.76.0 或更高版本。`aws-sdk-s3`

从命令行安装

对于安装了 `aws-sdk-s3` Gem 的项目，请使用版本选项来验证是否安装了 1.76.0 的最低版本。

```
gem install aws-sdk-s3 -v '>= 1.76.0'
```

使用 Gemfiles

对于使用 Gemfile 管理依赖关系的项目，请将 ge `aws-sdk-s3` m 的最低版本设置为 1.76.0。例如：

```
gem 'aws-sdk-s3', '>= 1.76.0'
```

1. 修改你的 Gemfile。
2. 运行 `bundle update aws-sdk-s3`。要验证您的版本，请运行`bundle info aws-sdk-s3`。

更新适用于 Ruby AWS 版本 2 的 SDK

适用于 Ruby 的 AWS SDK 第 2 版将于 2021 年 11 月 21 日进入维护模式。要完成此迁移，您需要使用 aws-sdk gem 的 2.11.562 或更高版本。

从命令行安装

对于安装了 `aws-sdk` gem 的项目，在命令行中使用版本选项来验证是否安装了 2.11.562 的最低版本。

```
gem install aws-sdk -v '>= 2.11.562'
```

使用 Gemfiles

对于使用 Gemfile 管理依赖关系的项目，请将 ge aws-sdk m 的最低版本设置为 2.11.562。例如：

```
gem 'aws-sdk', '>= 2.11.562'
```

1. 修改你的 Gemfile。如果你有 Gemfile.lock 文件，请将其删除或更新。
2. 运行 bundle update aws-sdk。要验证您的版本，请运行bundle info aws-sdk。

将加密和解密客户端迁移到 V2

更新客户端以读取新的加密格式后，您可以将应用程序更新到 V2 加密和解密客户端。以下步骤向您展示如何成功地将代码从 V1 迁移到 V2。

在更新代码以使用 V2 加密客户端之前，请确保您已执行上述步骤并使用的是 ge aws-sdk-s3 m 版本 2.11.562 或更高版本。

> **Note**
>
> 使用 AES-GCM 解密时，在开始使用解密的数据之前，请将整个对象读取到最后。这是为了验证该对象自加密以来未被修改。

配置 V2 加密客户端

EncryptionV2:: 客户端需要额外的配置。有关详细的配置信息，请参阅 EncryptionV2:: Client 文档或本主题后面提供的示例。

1. 在构造客户端时必须指定密钥封装方法和内容加密算法。创建新值时EncryptionV2::Client，需要为key_wrap_schema和提供值content_encryption_schema。

key_wrap_schema-如果您正在使用AWS KMS，则必须将其设置为:kms_context。如果您使用的是对称 (AES) 密钥，则必须将其设置为。:aes_gcm如果您使用的是非对称 (RSA) 密钥，则必须将其设置为。:rsa_oaep_sha1

content_encryption_schema-必须将其设置为:aes_gcm_no_padding。

2。 必须在客户端构造时指定 security_profile。创建新内容时EncryptionV2::Client，需要为提供值security_profile。security_pro file 参数决定了对读取使用较旧 V1 编写的对象的支持。Encryption::Client有两个值：: v2 和:v2_and_legacy。要支持迁移，请将设置为*security_profile*:v2_and_legacy。仅将:v2 用于新应用程序开发。

3。 AWS KMS key默认情况下，ID 是强制执行的。在 V1 中Encryption::Client，未向提供kms_key_id用于创建客户端的。AWS KMS Decrypt call AWS KMS可以从元数据中获取这些信息并将其添加到对称密文 blob 中。在 V2 中 e`ncryptionV2:: Client`，kms_key_id 被传递给解密调用，如果它与用于加密对象的AWS KMS密钥不匹配，则调用会失败。如果您的代码以前依赖于不设置特定kms_key_id值，则要么kms_key_id: :kms_allow_decrypt_with_any_cmk在创建客户端时设置，要么kms_allow_decrypt_with_any_cmk: true在get_object调用时设置。

示例：使用对称 (AES) 密钥

迁移前

```
client = Aws::S3::Encryption::Client.new(encryption_key: aes_key)
client.put_object(bucket: bucket, key: key, body: secret_data)
```

```
resp = client.get_object(bucket: bucket, key: key)
```

迁移后

```
client = Aws::S3::EncryptionV2::Client.new(
  encryption_key: rsa_key,
  key_wrap_schema: :rsa_oaep_sha1, # the key_wrap_schema must be rsa_oaep_sha1 for
 asymmetric keys
  content_encryption_schema: :aes_gcm_no_padding,
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by the
 V1 encryption client
 )
client.put_object(bucket: bucket, key: key, body: secret_data)  # No changes
resp = client.get_object(bucket: bucket, key: key) # No changes
```

示例：与 kms_key AWS KMS _id 一起使用

迁移前

```
client = Aws::S3::Encryption::Client.new(kms_key_id: kms_key_id)
client.put_object(bucket: bucket, key: key, body: secret_data)
resp = client.get_object(bucket: bucket, key: key)
```

迁移后

```
client = Aws::S3::EncryptionV2::Client.new(
  kms_key_id: kms_key_id,
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys
  content_encryption_schema: :aes_gcm_no_padding,
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by the
 V1 encryption client
)
client.put_object(bucket: bucket, key: key, body: secret_data)  # No changes
resp = client.get_object(bucket: bucket, key: key) # No change
```

示例：在没有 kms_ke AWS KMS y_id 的情况下使用

迁移前

```
client = Aws::S3::Encryption::Client.new(kms_key_id: kms_key_id)
client.put_object(bucket: bucket, key: key, body: secret_data)
resp = client.get_object(bucket: bucket, key: key)
```

迁移后

```
client = Aws::S3::EncryptionV2::Client.new(
  kms_key_id: kms_key_id,
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys
  content_encryption_schema: :aes_gcm_no_padding,
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by the
 V1 encryption client
)
client.put_object(bucket: bucket, key: key, body: secret_data)  # No changes
resp = client.get_object(bucket: bucket, key: key, kms_allow_decrypt_with_any_cmk: true) #
 To allow decrypting with any cmk
```

迁移后的替代方案

如果您仅使用 S2 加密客户端读取和解密（从不写入和加密）对象，请使用此代码。

```
client = Aws::S3::EncryptionV2::Client.new(
  kms_key_id: :kms_allow_decrypt_with_any_cmk, # set kms_key_id to allow all get_object
 requests to use any cmk
  key_wrap_schema: :kms_context, # the key_wrap_schema must be kms_context for KMS keys
  content_encryption_schema: :aes_gcm_no_padding,
  security_profile: :v2_and_legacy # to allow reading/decrypting objects encrypted by the
 V1 encryption client
)
resp = client.get_object(bucket: bucket, key: key) # No change
```

文档历史记录

下表介绍了本指南的重要更改。如需获取对此文档的更新的通知，您可以订阅 RSS 源。

变更	说明	日期
目录 (p. 314)	更新了目录，使代码示例更易于访问。	2023 年 6 月 1 日
IAM 最佳实践（更新）(p. 314)	更新了指南，使其符合 IAM 最佳实践。有关更多信息，请参阅 IAM 安全最佳实践。入门版更新。	2023 年 5 月 8 日
一般更新 (p. 314)	更新相关外部资源的欢迎页面。还更新了 v2.3 所需的最低 Ruby 版本。更新了AWS Key Management Service章节以反映术语更新。为清楚起见，更新了 REPL 实用程序的使用信息。	2022 年 8 月 8 日
更正损坏的链接 (p. 314)	修复了损坏的示例链接。删除了多余的提示和技巧页面；重定向到 Amazon EC2 示例内容。包括代码示例存储库GitHub中可用的代码示例列表。	2022 年 8 月 3 日
获取有关所有 Amazon RDS 安全组的信息 (p. 122)	添加了有关停用 EC2-Clac 的说明。	2022 年 7 月 26 日
开发工具包指标 (p. 314)	删除了有关为企业Support 启用 SDK 指标的信息，该信息已过时。	2022 年 1 月 28 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。